

Automatisches Generieren von Fingersatz für Klavierpartituren mithilfe von Machine-Learning

Joel Rixen

Masterarbeit • Studiengang Medieninformatik • Fachbereich Informatik und Medien • 20.08.2018

Motivation

Beim Erlernen des Klavierspiels kann es besonders für Neueinsteiger problematisch sein, einen guten Fingersatz zu finden. Das Ziel dieser Arbeit ist, ein Programm zu erstellen, was den Fingersatz für Klavierpartituren generieren kann.

Aufgabenstellung

Diese Aufgabe wurde über die letzten 20 Jahre mehrfach versucht zu lösen. Diese Lösungsansätze basieren meistens auf der gleichen Idee und funktionieren nur bei eher einfachen Klavierstücken. In dieser Arbeit wurde deswegen mit einem anderen Lösungsansatz (Machine-Learning) gearbeitet.

Bisherige Lösungsansätze

Die bisherigen Lösungen erstellen für ein Klavierstück einen Graph, der für jede Note jeden Fingersatz beinhaltet. Danach wird aufgrund von einer Kostenfunktion ausgewählt, welcher Fingersatz ausgewählt wird. Ein Beispiel ist in Abb. 1 zu sehen.



Abb. 1: Output eines generierten Fingersatzes mittels Kostenfunktion (nach Balliauw et al., 2015)

Konzeption

Einen Fingersatz zu generieren kann als Klassifizierungsproblem betrachtet werden, d.h. jeder Note wird eine Klasse (Finger) zugeordnet. Dabei spielt die Reihenfolge die wichtigste Rolle. Für solche Probleme eignen sich RNNs („recurrent neural networks“). Zum Lernen werden Datensätze benötigt, die in dieser Arbeit aus einem Trainingset (41.676 Noten) und Testset (13.017 Noten) bestehen. Diese Datensätze enthalten Klavierstücke im MusicXML-Format, für die manuell der Fingersatz bestimmt wurde. Jeder Note werden Eigenschaften entnommen (z.B. die Tonhöhe), die als Input für das neuronale Netz genutzt werden. Dazu müssen die MusicXML-Dateien in einem mehrstufigen automatisierten in CSV-Dateien übersetzt werden (s. Abb. 2).

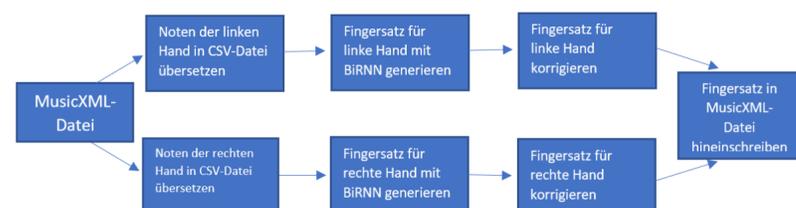


Abb. 2: Ablauf, um den Fingersatz zu generieren

Beim Training werden die Eigenschaften in das neuronale Netz eingelesen, verarbeitet und dann wird der Output mit dem vordefinierten Ergebnis verglichen. Wenn das Modell eine falsche Lösung liefert, so werden Werte („weights“, „biases“) angepasst. Nach dem Trainingsvorgang wird anhand des Testsets überprüft, wie gut das neuronale Netz verallgemeinern kann.

Umsetzung

Beim neuronalen Netz handelt es sich um ein BiRNN („bi-directional RNN“), welches die Sequenz der Inputs in beide Richtungen berücksichtigt. Die Klavierstücke im MusicXML-Format stammen von „musescore.com“. Als Machine-Learning-Framework wurde TensorFlow verwendet. Das BiRNN beinhaltet als Input eine 21 notenlange Sequenz und als Output fünf mögliche Klassen. Es werden Eigenschaften der zehn vorherigen und nachfolgenden Noten berücksichtigt bei der Bestimmung des Fingersatzes einer Note. Nach dem BiRNN wird mittels einiger Regeln der Fingersatz weiter verbessert.

Evaluation

Die Evaluation des Modells geschieht mittels Testsets. Ziel ist, herauszufinden, wie gut das Modell verallgemeinern kann. Da beim Fingersatz i.d.R. mehrere Lösungen möglich sind, müssen die generierten Fingersätze manuell getestet werden. Die Fehlerrate liegt bei 1% für die rechte und bei 2% für die linke Hand. Fast alle Fehler hängen mit einem inkorrekten Daumenuntersatz zusammen. Schlechter spielbare Fingersätze gab es bei beiden Händen zu ca. 3%. Es kommt also insgesamt zu 95% eine gute spielbare Lösung heraus (s. Abb. 3).



Abb. 3: Testperformance des Modells

Das Testset enthält Klavierstücke aus allen Schwierigkeitsstufen des G. Henle Verlags. Die Performance unterscheidet sich bei den verschiedenen Schwierigkeitsstufen nicht.

Fazit und Ausblick

Ein Beispiel eines generierten Fingersatzes ist in Abb. 4 zu sehen.



Abb. 4: Output eines generierten Fingersatzes des trainierten Modells

In dieser Arbeit wurde ein Lösungsansatz gefunden, der zu guten Ergebnissen bei allen Schwierigkeitsstufen von Klavierstücken führt. Die Fehlerrate könnte noch weit unter 1% verringert werden, wenn die Problematik des Daumenuntersatzes gelöst werden könnte.

Quellen

Balliauw, M., Herremans, D., Cuervo, D. P. & Sörensen, K. (2015). A Tabu Search Algorithm to Generate Piano Fingerings for Polyphonic Sheet Music. In T. Collins, D. Meredith & A. Volk (Eds.), *Mathematics and Computation in Music*. Berlin: Springer, 149-160.