Blues-Improvisation mit neuronalen Netze

Hagen Fritsch

<fritsch+blues@in.tum.de>

Seminar: Musik und Informatik

Technische Universität München

1. März 2007

Inhaltsverzeichnis

1 Einleitung

Das menschliche Gehirn übt seit jeher eine große Faszination auf Wissenschaftler vieler Forschungsrichtungen aus. Neben philosophischen Hintergründen über die Ursprünge der Menschheit fasziniert auch das Phänomen menschlicher Intelligenz, deren Ursprung und Funktionsweise bis heute nur unzureichend geklärt ist.

Seit der Entstehung der Informationstechnik beschäftigen sich Wissenschaftler mit der Frage, ob es Computern möglich ist, menschliche Intelligenz zu simulieren. In der Tat gibt es einige vielversprechende Ansätze, jedoch ist auch dieser Wissenschaftszweig immer noch Objekt intensiver Forschung. Einer dieser Ansätze sind die sogenannten künstlichen neuronalen Netze, deren Funktionsweise sich stark am biologischen Vorbild des menschlichen Gehirns orientiert. Dieser Ansatz bildet die wesentlichste Voraussetzung für diese Arbeit.

Wenn es darum geht, Intelligenz nachzubilden, ergibt sich automatisch die Frage, was denn Intelligenz eigentlich ist und was diese ausmacht. Einige Wissenschaftler sehen Kunst als höchste Form der Intelligenz, weswegen es auch von wissenschaftlichem Interesse ist, Computer zu Künstlern zu machen. Das hier vorgestellte Paper von Eck u. Schmidhuber (2002) widmet sich dem Versuch der Entwicklung einer Lernmethode, die es einem Computer ermöglichen soll, Blues-Musik zu improvisieren, also künstlerisch tätig zu werden.

Basis des Versuchs sind künstliche neuronale Netze, die im folgenden Kapitel in Grundzügen vorgestellt werden sollen. Im Anschluss daran, wird auf das konkrete Experiment detaillierter eingegangen und dabei Versuchsaufbau und Ergebnisse beschrieben.

2 Neuronale Netze

Künstliche neuronale Netze orientieren sich am biologischen Vorbild. Ein neuronales Netz ist ein Verbund aus Neuronen, die elektrische Impulse verschiedener Stärken austauschen. In den folgenden Abschnitten wird der Term „neuronales Netz“ oder einfach nur „Netz“ verwendet, meint aber immer das künstliche neuronale Netz.

2.1 Pattern Assoziator

Neuronale Netze werden in der Informatik z.B. als Ersatz für Funktionen gesehen, deren Berechnungsvorschrift unbekannt ist. Beim **supervised learning** soll das Netz anhand von Trainings- und bekannten Ausgabedaten *lernen*, wie die Eingabedaten im Netz verbreitet und manipuliert werden müssen, so dass die korrekten Ausgaben entstehen. Nach Abschluss der Lernphase hat das Netz das *Wissen*, einerseits die Trainingsdaten korrekt zu assoziieren, aber andererseits auch wahrscheinliche Vorhersagen für unbekannte Eingaben zu machen.

Das funktioniert allerdings nur, wenn das Netz eine klare Korrelation zwischen den Eingaben erkennen kann. Findet es keine, so muss es die Trainigsdaten sehr exakt lernen, um sie vorhersagen zu können. In dem Fall hat das Netz die Daten allerdings nur auswendig gelernt (Überanpassung, engl: *overfitting*). Vorhersagen für neue Eingabedaten sind dann nicht möglich.

2.2 Aufbau

Der Aufbau eines solchen künstlichen Netzes ist in Abbildung 1 illustriert. Die Kreise stellen die Neuronen (*Units*) dar. Die Verbindungen sind mit *Gewichten* versehen. Dem biologischen Vorbild entsprechend hat jedes Neuron ein Aktivitätslevel. Die Netzberechnung erfolgt nun von links nach rechts, wobei links die Eingabe- (*Input-*) und rechts die Ausgabeneuronen (*Output-Units*) liegen. Das Aktivitätslevel eines Neurons breitet sich nun entsprechend dem Gewicht der Verbindung an die jeweiligen Folgeneuronen aus.

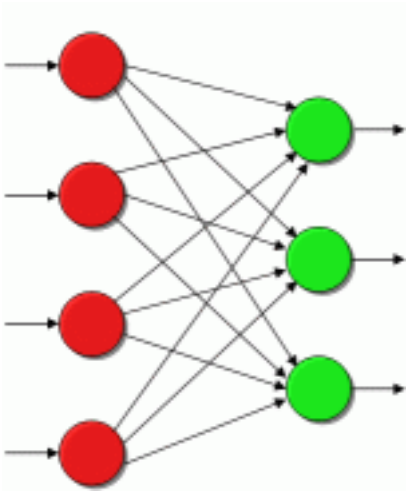

Abbildung 1: Modell eines einfachen neuronalen Netzes nach Rey (2006)

Jede Input-Unit hat einen Wert zwischen –1 und 1. Das Gewicht an einer Neuronenverbindung gibt nun an, welcher Anteil des Aktivitätslevels an die folgende Unit weitergegeben wird. Keinen physikalischen Grenzen unterworfen, kann ein Gewicht das Level aber auch vervielfachen (z.B. $w_{ij} = 2 = 200\%$). Die bei einer Unit eintreffenden Aktivitätslevel anderer Units werden addiert und bilden das neue Aktivitätslevel dieses Neurons.

2.3 Lernen

Das Wissen eines neuronalen Netzes ist in seinen Gewichten gespeichert. Damit es genutzt werden kann, muss es dem Netz zuerst trainiert werden. Training / Lernen bedeutet in diesem Kontext immer Gewichtsanpassung. Eine einheitliche Vorschrift, wie diese Gewichtsanpassungen vorgenommen werden müssen, existiert nicht, stattdessen gibt es mehrere Lernmethoden, von denen hier zwei beispielhaft genannt werden:

Die **Hebb-Regel** basiert auf der Beobachtung, dass die Verbindung zwischen zwei Neuronen gestärkt werden sollte, wenn beide gleichzeitig aktiv sind.

Die **Delta-Regel** vergleicht das gewünschte und tatsächlich erreichte Ak-

tivitätslevel eines Neurons und passt daraufhin die Gewichte aller eingehenden Verbindungen an, um eine bessere Annäherung an das gewünschte Aktivitätslevel zu erreichen.

2.4 Hidden Units

Die bisher vorgestellten neuronalen Netze haben lediglich zwei Schichten: Die Input- und die Output-Schicht. Da solch einfache Netze nicht sehr flexibel und lernfähig sind, wurden versteckte Zwischenschichten mit den sogenannten *Hidden Units* eingeführt. Diese sind praktisch Zwischenwerte auf dem Weg der Berechnung des Zielwertes. Anschaulich kann man sich eine Zwischenschicht als Modell der Wirklichkeit vorstellen. Ein Beispiel illustriert das ganz gut, wenn es auch nicht sehr realistisch gewählt ist, da sich neuronale Netze aufgrund der Komplexität des Beispiels nicht für die Realisierung eignen: Ziel des Netzes sei die Gesichtserkennung auf Fotos. Eingabedaten wären also die Pixel eines Bildes. Als Ausgabe wird der Name der Person erwartet, die auf dem Foto zu sehen ist. In einer Zwischenschicht könnte das Netz nun die Eingabedaten abstrahieren und beispielsweise Gesichtsmerkmale ableiten. Anhand dieser Merkmale könnte dann der Name assoziiert werden.

Das Problem versteckter Schichten ist offensichtlich, denn selbst beim *supervised learning* sind nun die Erwartungswerte der Zwischenschicht unbekannt. Doch auch dafür wurden Methoden entwickelt, wie der *Backpropagation*-Algorithmus, der ein Lernen mit Zwischenschichten ermöglicht. Beim diesem Algorithmus wird wie bei der Delta-Regel der Fehler an den Output-Units bestimmt und im Netz nun in entgegengesetzter Richtung bis zu den Input-Units ausgebreitet, wodurch Gewichtsanpassungen zu den *Hidden-Units* möglich werden.

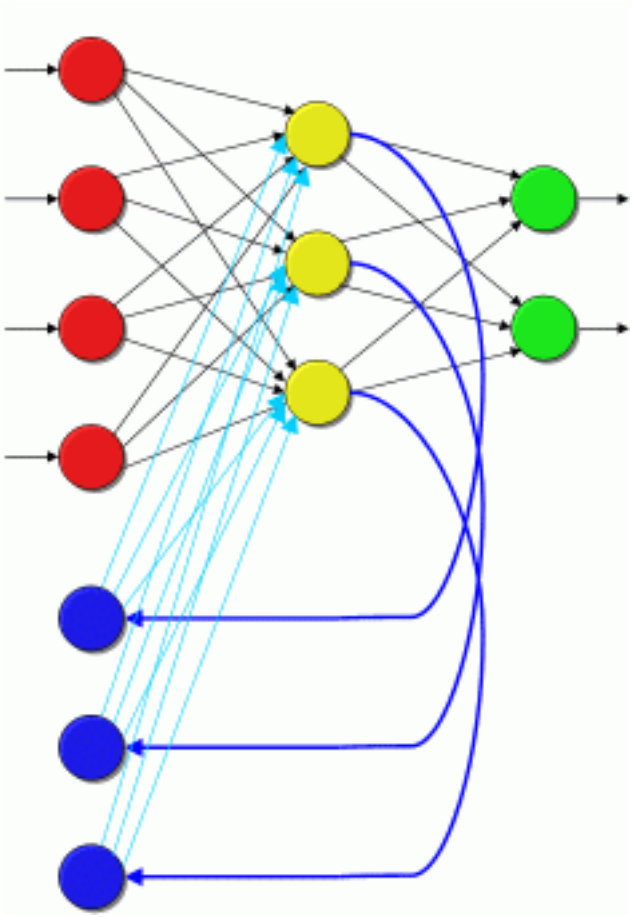

Abbildung 2: Illustration eines rekurrenten Netzes nach Rey (2006)

2.5 Erinnerungen

Netze mit Erinnerungen, sogenannte rekurrente Netze (engl.: *Recurrent Neuronal Network*: RNN) haben keine konsequente Links-Nach-Rechts-Berechnungsvorschrift mehr, sondern Quer- und Rückverbindungen, wie in Abbildung 2 illustriert. Das bewirkt, dass in die Berechnung eines Ausgabewertes, Berechnungen vorheriger Ausgabewerte einfließen. Das Netz „erinnert“ sich also.

Mit diesem mächtigen Mittel ausgestattet, könnte man nun bereits auf die Idee kommen, ein sich erinnerndes neuronales Netz zur Komposition von Musik zu verwenden. Es lernt anhand bestehender Melodien (den Trainings-

daten), welche Töne auf einen bestimmten gegebenen Ton folgen. Als sogenannter *Single-Step-Predictor*, sagt das Netz also den jeweils nächsten Ton voraus. Da es sich an die „Vergangenheit“ erinnern kann, ist die Voraussage nicht immer die selbe. Es kommt also Abwechslung in die Melodie.

In der Tat wurden ähnliche Experimente durchgeführt, mit bescheidenem Erfolg. Das Problem rekurrenter Netze ist, dass die Erinnerungen nicht langlebig sind und nach wenigen Zyklen (ca. 10) vom Netz bereits wieder vergessen sind. Eck u. Schmidhuber (2002) führen an, dass Musik einer globalen (zeitlichen) Struktur unterworfen ist. Langlebige Erinnerungen sind somit essentielle Voraussetzung für ein neuronales Netz mit dem Ziel der Musikimprovisation oder -komposition.

2.6 LSTM

Eck u. Schmidhuber bieten das *Long-Short-Term-Memory* (LSTM) als eine mögliche Lösung für dieses Problem an. Der Kern dieses LSTM sind LSTM-Zellen, die einige Neuronen ersetzen. Eine LSTM-Zelle ist nun ein komplexes Gebilde aus mehreren Neuronen, Quer- und Rückverbindungen, mit dem Ziel, relevante Informationen nicht zu vergessen, sondern lange zu erhalten.

3 Das Experiment: Blues Improvisation

Ziel des hier vorgestellten Experimentes ist es, zu zeigen, dass ein Computer Musikimprovisation erlernen kann, hier am Beispiel der Bebop-Blues-Improvisation. Insbesondere ging es Eck u. Schmidhuber auch darum, die Mächtigkeit von neuronalen Netzen mit LSTM-Zellen aufzuzeigen und zu illustrieren, dass gerade zeitliche Strukturen von solchen Zellen sehr gut erfasst werden.

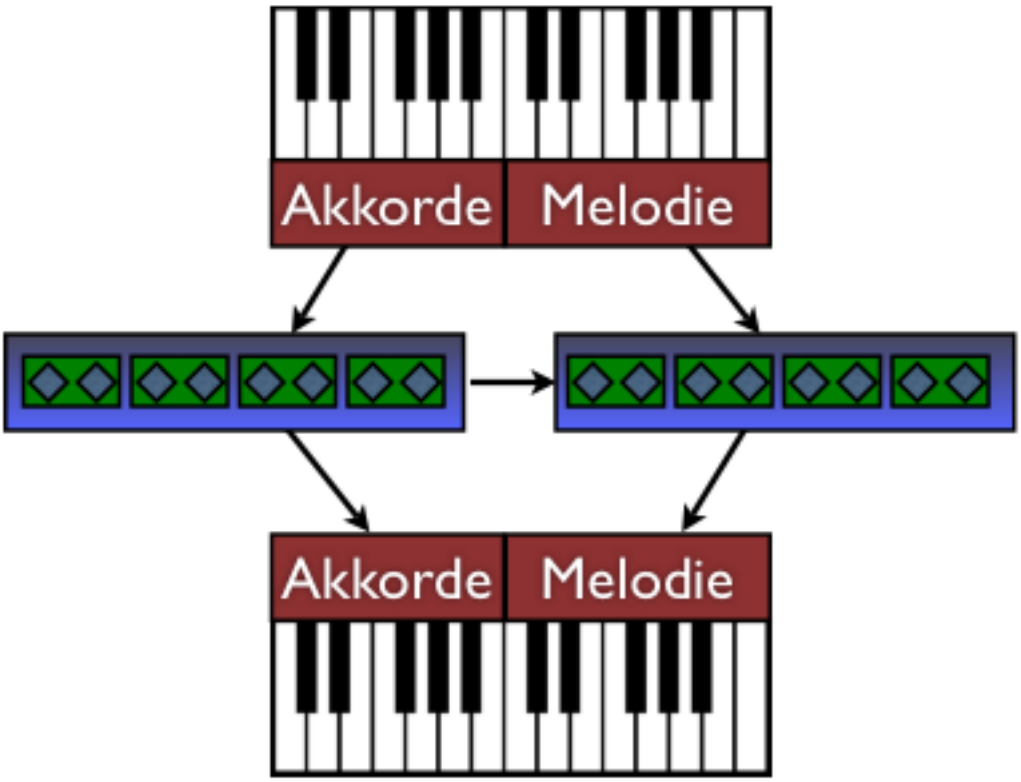

Abbildung 3: Versuchsaufbau

3.1 Versuchsaufbau

Blues-Musik zeichnet sich durch immer wiederkehrende, zeitlich konstante Akkordfolgen aus. Zu den Akkorden wird im Experiment frei improvisiert. An dieser Stelle sei darauf hingewiesen, dass in der Blues-Musik (wie auch bei den meisten anderen Formen von Musikimprovisation) die Melodie durch die Akkorde beeinflusst wird, nicht aber umgekehrt. Auf diesen Punkt wurde bereits im Versuchsaufbau großen Wert gelegt, sodass dieser Fakt beim Design des Netzes beachtet wurde.

Wie in Abbildung 3 illustriert, dienen dem Netz ein Akkord und ein Melodieton als Eingabe. Das Netz produziert dann den nächsten Ton als Ausgabe. Um die Komplexität einzuschränken, sind Melodie und Begleitung von ein-

ander getrennt und im Tonumfang eingeschränkt. Die Akkorde haben eine Oktave (12 Töne) zur Verfügung, die für jeden Akkord ausreichend ist, wobei einige zu diesem Zweck in einer Umkehrung gespielt werden. Die Melodie hat eine volle Oktave (12 Töne + oktavierten Grundton) zur Verfügung.

Der vom Netz „vorhergesagte“ Ton ist nun gleichzeitig wieder die Eingabe an das Netz. Somit ist das Netz von außen unabhängig und kann praktisch endlos weiter improvisieren.

3.2 Netzdesign und Timing

Wie vermittelt man einem solchen neuronalen Netz jetzt einzelne oder mehrere Noten, Pausen und Tonlängen? Es bieten sich einige Möglichkeiten. Eck u. Schmidhuber verwenden ein Eingabeneuron für jede Taste der Klaviatur. Dieses hat entweder das Aktivitätslevel 1.0, wenn die Taste gedrückt ist, oder 0.0, falls das nicht der Fall ist. Somit können beliebig viele und beliebig wenige Tasten gleichzeitig gedrückt sein. Um die Zeitkomponente mit ins Spiel zu bringen, lässt sich nun festlegen, dass die Eingaben beispielsweise, wie in den Experimenten des vorgestellten Papers, jede Achtelnote an das Netz geschickt werden. Somit wäre die kleinste Noteneinheit (Quantisierung) die Achtelnote. Eine längere Note wird dabei durch einen gedrückten Zustand über beispielsweise zwei Zeiteinheiten repräsentiert.

Noten	Achtel-Quantisierung	Sechzehntel-Quantisierung
	_ _ _ _	_ _ _ _ _ _ _ _
♩ ♩	_ _ _ _	_ _ _ _ _ _
♫ ♫	_ _ _ _	_ _ _ _

Tabelle 1: Notenrepräsentation bei unterschiedlicher Quantisierung

Für wechselnde Noten funktioniert dieser Ansatz einwandfrei. Wenn nun aber die selbe Achtelnote zweimal hintereinander gespielt würde, wäre dies, wie in Tabelle 1 illustriert, nicht von einer Viertelnote zu unterscheiden, die dann eine Länge von zwei Achtelnoten hätte. Eck u. Schmidhuber stört dieser

Fakt für das Experiment nicht. Sie führen jedoch an, dass man um diesen Effekt zu vermeiden, das Quantisierungs-Intervall verkleinern müsse. Somit könne man nach jeder „normalen“ Eingabe an das Netz eine „Halteeingabe“ feuern, die besagt ob die Note zu Ende ist, oder weiter gehalten wird.

Abbildung 3 illustriert noch einmal den Aufbau. Das Netz selbst besteht also aus acht Zellblöcken mit je zwei LSTM-Zellen. Davon sind jeweils vier für Akkorde und Melodie zuständig. Wie bereits erläutert und in der Abbildung ersichtlich, haben die Akkord-Zellblöcke Verbindungen zu denen der Melodie (und sich selbst), jedoch nicht umgekehrt.

3.3 Trainingsdaten

Für das im Paper beschriebene Experiment wurden als Trainingsdaten eine Bepop-Akkordfolge und eine selbstkomponierte Melodie zugrunde gelegt. Die Melodie wiederholt sich sehr oft, jedoch mit jeweils kleinen Veränderungen. Dieses Design wurde gewählt, um dem Netz einerseits Veränderungsspielraum zu lassen und andererseits eine gewisse harmonische Basis zu bieten, um dem Netz sein Möglichkeiten beizubringen. Als Basis für die Melodie diente eine pentatonische Skala (Fünftonleiter), die in der Blues-Musik die einfachste Grundlage für jegliche Improvisationen bildet.

3.4 Ergebnisse

Um wirklich zu sehen, dass das Netz mehr leistet, als eine einfache und zufällige Verteilung von Noten der pentatonischen Skala, wurde ein nach diesem Prinzip „komponiertes“ Stück zum Vergleich dargeboten. Auch wenn es sich bereits nach Blues-Musik anhört (bei einer pentatonischen Skala kann man grundsätzlich mit keiner Note verkehrt liegen), so fehlt es doch an jeglicher melodischer Struktur.

Die Akkordfolge der Musik wird im Experiment wie erwartet vorhergesagt. Anhand der Beispiele zeigt sich schließlich aber auch, dass sich die Melodie an den Akkorden ausrichtet. Melodisch „klingen“ die Ergebnisse

wesentlich strukturierter und gelegentlich lassen sich Melodieteile des Trainingsthemas erkennen. Eck u. Schmidhuber beschreiben den sehr einleuchtenden Sachverhalt, dass die Länge der Lernphase ausschlaggebend ist für die Präsenz des Themas innerhalb der „improvisierten“ Melodie. Während eine kurze Lernphase in wenig zusammenhängen Themen resultiert und dabei die Tonwahl eher zufällig wirkt, werden die erzeugten Themen bei längeren Lernphasen verstärkt Ähnlichkeit zum Trainingsthema aufweisen, eine Befund der sich auch beim menschlichen Lernen findet: Je öfter man etwas wiederholt, desto besser prägt man es sich ein.

3.5 Zusammenfassung

Im Experiment konnte gezeigt werden, dass es einem neuronalen Netz gelingt, logische und zeitliche Strukturen anhand von Trainingsmelodien und -akkorden zu erlernen und daraus selbst neue, harmonisch stimmige Blues-Musik des selben Stils zu „improvisieren“.

Wenngleich dieses Experiment erfolgreich war, so ist noch lange nicht gezeigt, dass es Computern möglich ist, die komplexe Intelligenz eines Menschen zu erlangen. Bisherige Versuche zur Musikkomposition mit neuronalen Netzen konnten nur bescheidenen Erfolg aufweisen. Die Lösung durch Verwendung von LSTM-Zellen mag in diesem Fall geholfen haben, entfernt sich jedoch stark vom biologischen Vorbild, weshalb gerade die Zukunft der LSTM-Zellen ungewiss ist und die Suche nach besseren Modellen und Lernmethoden weitergeht.

Literatur

Eck u. Schmidhuber 2002

Eck, Douglas ; Schmidhuber, Jürgen: Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. Version: 2002. `http://www.idsia.ch/~juergen/blues/`. In: *Bourlard, H., ed.: Neural Networks for Signal Processing XII, Proceedings of the 2002 IEEE Workshop, New York, IEEE.* 2002, 747–756

Rey 2006

Rey, Günter D.: *Neuronale Netze: Eine Einführung.* `http://neuronalesnetz.de/downloads/nn_druckversion.pdf`. Version: 2006, Abruf: 1. März 2007