# Zelluläre Automaten

Jan Köhler

24. Juni 2011

## Grundlagen

Zelluläre Automaten werden zur Beschreibung zeitlich und räumlich diskreter Systeme verwendet. Der Zustand einzelner Zellen zum Zeitpunkt $t + 1$ wird dabei meist aus dem Zustand der Nachbarzellen zum Zeitpunkt $t$ bestimmt.
Typische Anwendungsbeispiele für Zelluläre Automaten sind:

- Neuronale Netze. Nervenzellen werden als 2D Gitter dargestellt, eine Nervenzelle wird aktiv wenn im vorigem Zeitschritt ein Nachbar aktiv war.

- Simulation des Straßenverkehrs. Einzelne Zellen repräsentieren Fahrzeuge, zu jedem Zeitschritt bewegt sich ein Fahrzeug einige Zelle in Richtung seines Ziels. Mit einfachen Regeln wie bremsen, trödeln, fahren, beschleunigen war es zum ersten mal möglich den Straßenverkehr realistisch zu beschreiben und Effekte wie den "Stau aus dem Nichts" zu erklären.

- *Conway's Game of Life*. Mit einem Satz einfacher Regeln entstehen auf einem 2D Gitter komplexe Strukturen wie Gleiter, Blinker, Segler ... und andere stabile, oszillierende oder sogar sich selbst replizierende Objekte.

- Das *Ising-Modell* beschreibt den Ferromagnetismus in Festkörpern. Die magnetischen Momente der Atome angeordnet werden auf einem Gitter angeordnet, in Abhängigkeit der vorgegebenen Temperatur bilden sich verschiedene Strukturen, von perfekter Ordung bis zu perfektem Rauschen, die verschiedenen Magnetisierungen entsprechen.

## Aufgabenstellung

Simulieren sie das Game of Life und das Ising-Modell auf einem 2D Gitter und stellen sie beide grafisch dar. Die Berechnung der Automaten auf einem 2D Gitter soll möglichst allgemein gelöst werden, d. h. eine Basisklasse beschreibt die zeitliche Entwicklung des Gitters, die verschiedenen Modelle sollten als abgeleitete Klassen realisiert werden. Implementieren Sie Methoden, um Anfangszustände ($t = 0$) zu laden und Ergebnisse zu speichern.

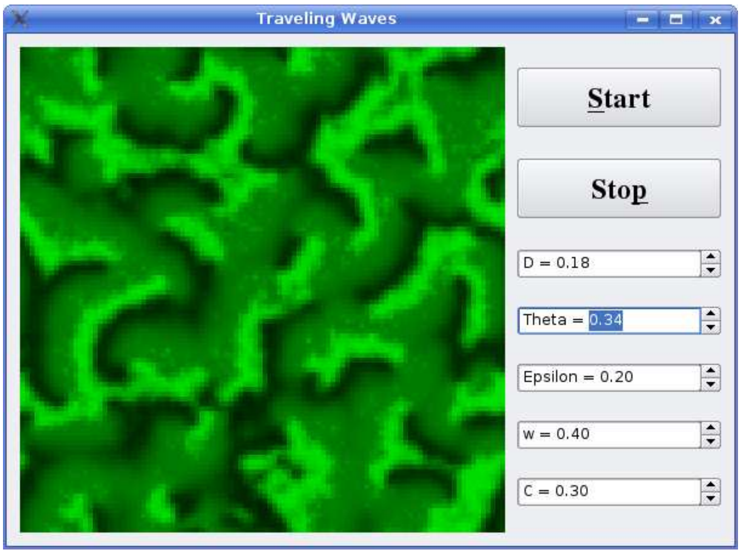


Abbildung 1: Neuronale Aktivität (grün) breitet sich auf einem 2D Gitter aus. Die verschiedenen Parameter können während der Simulation verändert werden.

## Vorgehen

1. Erstellen sie die Klasse `automat` die folgende Elemente enthält:
   - Das Gitter als 2D `numpy` array
   - Eine leere Funktion, die den Zustand einer Zelle im nächsten Zeitschritt berechnet.
   - Eine Funktion, die den Zustand aller Zellen im nächsten Zeitschritt berechnet.
   - Funktionen zum laden und speichern ...
   - Eine Methode um $N$ Schritte mit `Gnuplot` zu visualisieren. *Hinweis: Um die Geschwindigkeit zu regeln, ist es sinnvoll, nach jedem Zeitschritt einige Millisekunden zu warten (time.sleep).*
2. Benutzen sie das Konzept der Vererbung um von `automat` die Klasse `GameOfLife` abzuleiten.
3. Benutzen sie das Konzept der Vererbung um von `automat` die Klasse `IsingModell` abzuleiten.

Optional Benutzen sie `pyQT` um eine erweiterte Benutzeroberfläche zu erstellen, bei der die Simulation angehalten, abgespeichert, fortgesetzt werden kann und bei der die Parameter des Ising Modells zur Laufzeit geändert werden können vgl. (Abb. 1).

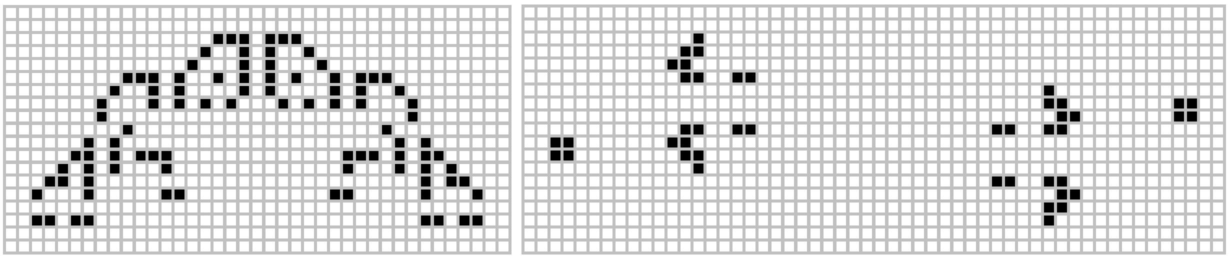

Abbildung 2: Zwei Muster die im Game of Life auftreten können. Links: Das Barge 2 Spaceship bewegt sich in jedem Zeitschritt ein Felde nach oben. Rechts: Bei der Bi-gun wandern zwei sog. twin bee shuttles zwischen zwei Quadern hin und her und erzeugen bei jeder Kollision einen Gleiter.

# Game of Life

Das Game of Life beschreibt die Evolution von Zellen auf einem 2D Gitter. Zur Berechnung des nächsten Zeitschritts werden vier Regeln angewendet:

- Eine tote Zelle mit genau drei lebenden Nachbarn wird im nächsten Zeitschritt neu geboren.
- Lebende Zellen mit weniger als zwei lebenden Nachbarn sterben an Einsamkeit.
- Eine lebende Zelle mit zwei oder drei lebenden Nachbarn bleibt lebend.
- Lebende Zellen mit mehr als drei lebenden Nachbarn sterben an Überbevölkerung.

Eine ausführliche Beschreibung ist unter [1] zu finden. Abb. 2 zeigt zwei Beispiele von auftretenden Strukturen. Eine Vielzahl von Beispielen ist unter [2] zu finden.

# Ising-Modell

Das Ising-Modell beschreibt den Magnetismus in Festkörpern in einem vereinfachtem Modell in dem das magnetische Moment einzelner Atome die Spinwerte $S = \pm 1$ annimmt. Es wird durch den Hamiltonoperator

$$\mathcal{H} = -\frac{1}{2} \sum_{i,j} J_{i,j} S_i S_j$$

beschrieben, wobei die $S_i$ die einzelnen Spins darstellen und $J_{i,j}$ die Kopplung zwischen Spins am Ort $i$ und $j$. Für ferromagnetische Kopplungen ist $J_{i,j} \geq 0$, für antiferromagnetische Kopplungen ist $J_{i,j} \leq 0$. Offensichtlich ist die Energie für eine ferromagnetische Kopplung minimal wenn alle Spins die gleiche Ausrichtung haben. Diese magnetische Ordnung herrscht jedoch nur bei tiefen Temperaturen, bei höheren Temperaturen wird die Ordnung durch Spinfluktuationen aufgebrochen wobei ein Phasenübergang stattfindet. Weiterführende Informationen sind unter [3] zu finden.

Zur Simulation in einem zellulären Automaten betrachten wir das Ising-Modell in zwei Dimensionen und betrachten die vereinfachte Kopplung $J_{i,j} = 1$ für die vier benachbarten Spins und $J_{i,j} = 0$ sonst.
Je nach Temperatur existiert eine Wahrscheinlichkeit, dass sich ein Spin gleich oder entgegengesetzt zu seinen Nachbarn ausrichtet. Nach dem Wärmebad Algorithmus ist die Wahrscheinlichkeit, dass sich ein Spin $S_i$ im folgendem Zeitschritt im Zustand $\pm 1$ befindet durch

$$W(S_i = \pm 1) = \frac{e^{\pm E_i/T}}{e^{-E_i/T} + e^{+E_i/T}}$$

gegeben, wobei $E_i$ in diesem Fall die Summe der vier benachbarten Spins und $T$ die dimensionslose Temperatur ist. Verschiedene Möglichkeiten zur Simulation des Ising-Modells werden in [4] erläutert.

## Vorschläge

- Erweitern sie die Simulation um ein Externes Magnetfeld
- Vergleichen sie den Wärmebad Algorithmus mit dem Metropolis Algorithmus

# Literatur


[1] http://de.wikipedia.org/wiki/Conways_Spiel_des_Lebens

[2] http://www.conwaylife.com/wiki/

[3] http://de.wikipedia.org/wiki/Ising-Modell

[4] http://pha.physik.hu-berlin.de/lehre/SemTPMB08/vortrag_rheinwalt.pdf