Chapter 8

Neuronale Netze und abstrakte Automaten

Sowohl die bisher besprochenen Feed–Forward–Netze als auch die rekurrenten Netze können als spezielle abstrakte Automaten aufgefaßt werden. Auch die Attraktornetzwerke stehen in enger Verwandschaft zu endlichen Automaten und sind insbesondere Verallgemeinerungen von zellulären Automaten.

8.1 Abstrakte Automaten

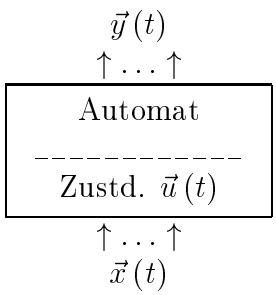
Ein abstrakter Automat ist ein Tupel

$$\mathbf{A} = (\mathbf{X}, \mathbf{Y}, \mathbf{U}, f, g) \tag{8.1}$$

mit

- $\mathbf{X}$ = Eingabealphabet, mit den Elementen $x_i \in \mathbf{X}$, die Eingabebuchstaben genannt werden. Analog ist $\mathbf{Y}$ das Ausgabealphabet.
- $\mathbf{U}$ ist die Menge der inneren Zustände.
- Die Funktionen f bzw. g vermitteln eindeutige Abbildungen der Produktmengen $\mathbf{X} \times \mathbf{U}$ in die Menge der Zustände $\mathbf{U}$ bzw. in die Ausgabemenge $\mathbf{Y}$. Diese Abbildungen definieren die Übergänge zwischen den Zuständen bzw. die Ausgabe des Automaten als Funktion der Eingabe und des aktuellen inneren Zustandes von A, s.u. Folglich heißt f auch die Überführungs– und g die Ergebnisfunktion des Automaten.

Die Arbeit des Automaten



(8.2)

ist bezüglich der getakteten Zeit $t = 0, 1, ...$ durch folgenden Algorithmus bestimmt:

1. Initialisierung. Zum Zeitpunkt (o.B.d.A.) $t = 0$: Einlesen des inneren Zustandes (Startzustand) $\vec{u}$ des Automaten.

2. Einlesen eines Buchstabens $\vec{x}$ aus dem Eingabealphabet. Bestimmung der Ausgabe $\vec{y}\,(t)$ und des Folgezustandes $\vec{u}\,(t+1) = \vec{u}^{neu}$ seines aktuellen inneren Zustandes $\vec{u}$

$$\begin{aligned} \vec{y}\,(t) &= g\,(\vec{x}, \vec{u}) \\ \vec{u}\,(t+1) &= f\,(\vec{x}, \vec{u}) \end{aligned} \tag{8.3}$$

3. $\vec{u} = \vec{u}^{neu}$ ($\vec{u}^{neu}$ wird neuer aktueller innerer Zustand). $t = t + 1$

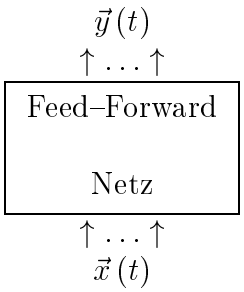
4. Abbruch? *GOTO* 2

Als endliche Automaten werden speziell solche bezeichnet, für die die Mengen $\mathbf{X}, \mathbf{Y}, \mathbf{U}$ endlich sind.

8.2 Neuronale Netze als abstrakte Automaten

Im folgenden wollen wir den formalen Zusammenhang zwischen verschiedenen neuronalen Netzen und abstrakten Automaten darstellen.

- Die in Kap. **??** besprochenen Feed–Forward–Netze



(8.4)

  sind spezielle Automaten, die sich von 8.2 einzig dadurch unterscheiden, daß sie nicht über einen inneren Zustand verfügen, der die Ausgabe in Abhängigkeit von den Inputs $\vec{x}$ beeinflussen könnte. Die Buchstaben des Ein– bzw. Ausgabealphabetes des Automaten sind mit der Menge aller möglichen Input– bzw. Outputvektoren des Netzes zu identifizieren. Feed–Forward–Netze können wegen dieser Eigenschaft als ein im Sinne der Signaltheorie kombinatorische Systeme bezeichnet werden.

- **Rekurrente Netze:** Der fehlende innere Zustand der Feed–Forward–Netze macht den Output des Netzes zu einer direkten Funktion des aktuellen Inputs des Netzes. Auf diese Weise geht die Vorgeschichte der Inputs nicht in den Ouput ein. Das ist vor allem für die Verwendung von Netzen als Modelle von Zeitreihen oder allgemeiner von dynamischen Systemen oft ein entscheidender Nachteil.
  Dieses Problem kann durch die Verwendung rekurrenter Netze beseitigt werden. In rekurrenten Netzen koppeln ausgewählte Neuronen ihren Output auf sich selbst oder auf andere Neuronen zurück. Die Outputs dieser ausgewählten Neuronen können als innerer Zustand des Netzes $\vec{u}$ interpretiert werden, der nach der Regel 8.3 upgedatet wird. Damit sind rekurrente Netze Realisierungen "echter" abstrakter Automaten und können wie diese eine innere Repräsentation der Vorgeschichte des aktuellen Inputs aufbauen.
  Für die Beziehungen zwischen rekurrenten neuronalen Netzen und abstrakten Automaten existieren eine Reihe von strengen Aussagen. So wurde schon 1991 von Alon, Dewdney und Ott [1] gezeigt, daß rekurrente neuronale Netze jeden endlichen abstrakten Automaten in Echtzeit simulieren können. Siegelman und Sontag [4] bewiesen 1995

die gleiche Aussage für die Simulation einer multi–stack Turing Maschine. Damit besitzen die rekurrenten Netze die gleiche Mächtigkeit wie diese Berechnungsmodelle.

Diese Aussagen sind zunächst nur recht prinzipieller Natur, es konnten aber auch Ergebnisse für Netze konkreter Architekturen gewonnen werden. So gilt z. B. der Satz, daß sich rekurrente Netze vom Typ der von Elman, vgl. [2, 3] zuerst eingeführten Architektur zur Simulation eines beliebigen endlichen Automaten in Echtzeit eignen, während rekurrente Kaskadenkorrelationsnetze diese Performanz nicht haben, vgl. [5].

- **Attraktornetzwerke (Hopfieldnetz):** Eine weitere Spielart der abstrakten Automaten sind die in **??** dargestellten Attraktornetzwerke (Hopfielnetzwerk). Diese dienen als Assoziativspeicher, die einen beliebigen Inputvektor $\vec{x}$ auf einen der gespeicherten Mustervektoren $\vec{x}^{\mu}$ abbilden. Diese Abbildung ist ein Ergebnis der Attraktordynamik. Das Netz wird mit $\vec{x}$ als Anfangszustand gestartet und erreicht nach endlicher Zeit einen Fixpunkt, der die Ausgabe des Speichers darstellt. Das entspricht einem Automaten mit Ergebnis– bzw. Überführungsfunktionen, die nicht von einem Eingabebuchstaben abhängen, d. h.

$$\vec{y}(t) = g(\vec{u}) \tag{8.5}$$

$$\vec{u}(t+1) = f(\vec{u})$$

anstelle von 8.3.

Beim Hopfieldnetz ist speziell $g(\vec{u}) = \vec{u}$. Die Eingabe erfolgt über die Initialisierung des inneren Zustandes, d. h. $\vec{u}_0 = \vec{x}$. Die Überführungsfunktion ist so zu wählen, daß der Automat nach einer endlichen Zahl von Schritten zum Halten kommt. Die Ausgabe des Assoziativspeichers ist mit $\vec{y} = g\left(\vec{u}^{halt}\right)$ durch den inneren Zustand $\vec{u}^{halt}$ des Automaten gegeben. Wegen 8.5 sind die Attraktornetze spezielle Realisierungen des Moore–Automaten.

8.3 Stochastische Automaten und Netze

In Vorbereitung

Bibliography

[1] N. Alon, A. K. Dewdney, and T. Ott. Efficient simulation of finite automata by neural nets. *Journal of the Association for Computing Machinery*, 38:495 – 514, 1991.

[2] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179 – 211, 1990.

[3] J. L. Elman. Language as a dynamical system. In R. F. Port and T. V. Gelder, editors, *Mind as Motion: Explorations in the Dynamics of Cognition*, chapter 8, pages 195 – 226. MIT, 1995.

[4] H. T. Siegelman and E. D. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50:132 – 150, 1995.

[5] A. Sperduti. On the computational power of recurrent neural networks for structures. *Neural Networks*, 10(3):395 – 400, 1997.