Zusammenfassung der Dissertation:

Neural Networks and the Complexity of Languages

Neuronale Netze und Sprachkomplexität

André Grüning
Juli 2003

1 Einführung

Die Frage nach der Natur menschlicher Sprache ist eng verknüpft mit den Arbeiten von Noam Chomsky [2, 3] auf den Gebieten der Linguistik und Informatik. Im Rahmen der von Chomsky begründeten *Generativen Grammatik* wird natürliche Sprache durch zumeist kontext-freie Grundgrammatiken [9] beschrieben. Der große Erfolg dieser Theorie liegt darin, daß sie sehr viele auf den ersten Blick stark verschiedene natürliche Sprachen auf gemeinsame zugrundeliegende Prinzipien zurückführen kann. Die Generative Grammatik versteht sich selbst als eine kognitive Theorie und geht davon aus, daß diese Prinzipien dem Menschen angeboren sind, da andernfalls Sprache nicht erlernbar sei [7]. Es stellt sich jedoch heraus, daß gerade die kognitiven Annahmen der Generativen Grammatik problematisch sind.

Künstliche neuronale Netze sind biologisch inspirierte Modelle der Architektur des Gehirns. Im Rahmen des Konnektionismus [6] werden sie verwendet, um kognitive Prozesse zu modellieren. Dem Konnektionismus liegt die Annahme zugrunde, daß kognitive Prozesse besser zu modellieren sind durch subsymbolische Operationen einer großen Anzahl einfacher Elementarprozessoren (den Neuronen) als durch symbolische Regelsysteme. Konnektionistische Modelle konnten zeigen, daß linguistische Strukturen von neuronalen Netzen gelernt werden können, auch wenn sie kein sprachliches Vorwissen in Form von angeborenen Prinzipien im Sinne der Generativen Grammatik haben.

Neuronale Netze (genauer: sog. *simple recurrent networks*) [5] lösen die Aufgabe, eine Sprache zu verarbeiten, in dem sie ein einfaches dynamisches System verkörpern [10]. Sie gleichen also eher analogen als symbolischen oder digitalen Computern.

In dieser Arbeit tragen wir zu folgenden Fragen bei: Wie werden gewisse typische sprachliche Strukturen in neuronalen Netzen repräsentiert? Welche Strukturen sind einfach aus der Sicht eines neuronalen Netzes? Sind diese einfachen Strukturen ähnlich denen, die häufig in natürlichen Sprachen vorkommen? Und können wir letztendlich diese einfachen Strukturen wieder durch wohldefinierte Komplexitätsmaße beschreiben?

2 Grundwortstellung

Viele Sprachen haben eine sog. Grundwortstellung, die Reihenfolge, in der Subjekt (`S`), Verb (`V`) und Objekt (`O`) in einem einfachen Aussagesatz erscheinen:

Englisch: *The girl* *read* *the newspaper.*
`S` `V` `O`

Walisisch: *Lladdodd* *y ddraig* *y dyn.*
tötete der Drachen den Mann
V S O

'Der Drachen tötete den Mann.'

Japanisch: *Gakusei-ga* *hon-o* *yonda.*
Student Buch las
S O V

'Der Student las ein/das Buch.'

In einer Reihe von Simulationen trainieren wir neuronale Netze auf ein kleines Korpus einer Modellsprache, das in verschiedenen Grundwortstellungen ausgegeben wurde, mit dem Ergebnis, daß die SOV-Sprache etwas besser als die SVO-Sprache und beide wesentlich besser als die VSO-Sprache gelernt wurden. Entropiebetrachtungen und Größe des minimalen endlichen Automaten als von neuronalen Netzen unabhängige Komplexitätsmaße führen zu demselben Resultat, daß insbesondere VSO schwieriger zu lernen ist.

Darüber hinaus reflektieren die Simulationsergebnisse die Häufigkeiten, mit denen die Grundwortstellungen SOV (45%), SVO (42%), VSO (9%) unter den natürlichen Sprachen vorkommen. Legt man ein evolutionäres Bild der Sprachentwicklung zugrunde, so sind sowohl für Menschen als auch für neuronale Netze die gleichen Wordreihenfolgen einfach.

3 Generische kontext-freie und kontext-sensitive Sprachen

In natürlichen Sprachen tauchen häufig verschachtelte Strukturen der Form *abccba* auf. Die zugrundeliegende Struktur ist die eines Palindroms ww^r, wobei w eine beliebige Zeichenkette und w^r ihr Spiegelbild ist. Sehr viel seltener tauchen Strukturen der Form *abcabc* auf, die als Duplikationssprache ww abstrahiert werden können.

Von einem symbolischen Standpunkt ist ww^r einfacher als ww, denn ww^r hat eine kontext-freie Grammatik, während man für ww kontext-sensitive Regeln benötigt. Für analoge Computer gilt dies nicht a priori. In einer Serie von Simulationen haben wir deswegen neuronale Netze entweder auf ww^r oder ww mit w über einem binären Alphabet $\{0, 1\}$ trainiert. Die Simulationsergebnisse zeigen, daß ww schwieriger zu erlernen ist als ww^r.

Der Grund dafür liegt in den einfachen dynamischen Systemen, die während des Trainings in den Netzen entstehen. Beide Sprachen können im wesentlichen in zweidimensionalen dynamischen Systemen verarbeitet werden, und die idealisierten dynamischen Systeme sind wie folgt gegeben:

Palindrom ww^r:

$$\begin{aligned}
\text{Ausgangszustand}: &\ x := 0, y := 1 \\
\textsc{push}(0): &\ f_0 : (x, y) \mapsto (\frac{1}{2}x, \frac{1}{2}y) \\
\textsc{push}(1): &\ f_1 : (x, y) \mapsto (\frac{1}{2}x + \frac{1}{2}, \frac{1}{2}y) \\
\textsc{pop}(): &\ f_{\P} : (x, y) \mapsto (2x \mod 1, 2y)
\end{aligned}$$

Duplikations ww:

$$\begin{aligned}\text{Ausgangszustand}: \; & x := 0, y := 1 \\ \text{PUSH}(0): \; & f_0 : (x, y) \mapsto (x, \frac{1}{2}y) \\ \text{PUSH}(1): \; & f_1 : (x, y) \mapsto (x + \frac{1}{2}y, \frac{1}{2}y) \\ \text{POP}(): \; & f_{¶} : (x, y) \mapsto (2x \mod 1, 2y)\end{aligned}$$

Die binäre Expansion von $x = 0.x_1x_2\cdots$ arbeitet wie ein Kellerspeicher bzw. wie eine Warteschlange: Nach der Verarbeitung von initialem w durch f_0 und f_1 stimmt die binäre Darstellung von $x = 0.x_1x_2\cdots$ mit dem Spiegelbild von w für ww^r bzw. mit w selbst für ww überein. Während der Verarbeitung der zweiten Hälfte w^r bzw. w durch $f_{¶}$ muß lediglich die Operation $x \to 2x \mod 1$ ausgeführt werden und das jeweilige Symbol in w mit der am meisten signifikanten Stelle von x verglichen werden. y arbeitet wie ein multiplikativer Zähler, der leeren Keller bzw. leere Warteschlange anzeigen kann.

Der Grund dafür, daß ww auch in neuronalen Netzen schwerer zu repräsentieren ist, liegt daran, daß die Dynamiken von x und y im Gegensatz zu ww^r nicht unabhängig sind, sondern x durch f_1 in Abhängigkeit von y manipuliert wird.

Zusammenfassend haben wir die Natur der Repräsentation von ww^r und ww in neuronalen Netzen aufgeklärt und eine Begründung für ihre unterschiedliche Schwierigkeit oder Komplexität gefunden. Falls neuronale Netze ein gutes Modell für kognitive Prozesse sind, liefert uns dies auch eine Erklärung, warum Strukturen wie ww in natürlichen Sprachen selten vorkommen.

4 Blockentropien und reguläre Sprachen

In diesem Teil der Arbeit unternehmen wir erste Schritte, Komplexitätmaße zu finden, die für das Sprachlernen von neuronalen Netzen relevant sind. Wir wollen zeigen, daß reguläre Sprachen, die ja die einfachste Sprachklasse in der Chomsky-Hierarchie darstellen, auch unter informationstheoretischen Gesichtspunkten einfach sind [4].

Ein endlicher Automat mit Zustandsmenge S kann als Quelle eines stochastischen Prozesses angesehen werden, wenn man den ausgehenden Übergängen eines jeden Knotens Wahrscheinlichkeiten zuordnet, mit denen sie benutzt werden. Ist ein solcher Automat irreduzibel, d. h. gibt es von jedem Zustand einen endlichen Pfad zu jedem anderen Zustand, und aperiodisch, d. h. kann man sich für eine große Anzahl von Schritten unabhängig vom Ausgangszustand im Prinzip in einem beliebigen Zustand befinden, so kann dieser Automat einen stationären stochastischen Prozeß darstellen. Dieser Prozeß erzeugt kontinuierlich Wörter aus einer regulären Sprache über einem Alphabet A. Ein Maß für die Schwierigkeit, das nächste Symbol $a \in A$ bei einem solchen Prozeß aus den vorangehenden n Symbolen vorherzusagen, sind die *n-Blockentropien* $H(A|A^n)$. Es ist bekannt, daß für einen stationären stochastischen Prozeß $\lim_{n\to\infty} H(A|A^n)$ gegen die sog. Entropierate h konvergiert. Die Geschwindigkeit der Konvergenz wird verschiedentlich als ein gutes Maß für die Komplexität eines stochastischen Prozesses angenommen [8, 1]. Wir können zeigen:

Satz 4.1 *Sei $\mathcal{S}$ der stationäre Prozeß, der von einem irreduziblen und aperiodischen endlichen Automaten, einer dazu kompatiblen stochastischen Matrix P und deren stationären Vektor p_S*

erzeugt wird. Dann gilt für geeignete Konstanten $0 \leq c_1 < 1$ *und* $c_0 \leq 0$*:*

$$H(A|A^n) - H(A|S) \leq c_0 c_1^n \tag{1}$$

und deshalb auch

$$h = \lim_{n\to\infty} H(A|A^n) = H(A|S). \tag{2}$$

$H(A|S)$ bezeichnet dabei die gemittelte Information, die nötig ist, aus der Kenntnis des momentanen Zustands $s \in S$ des endlichen Automaten das nächste Symbol $a \in A$ zu bestimmen. Wesentlich in den Beweis geht ein, daß es für jeden endlichen Automaten ein *synchronisierendes Word* gibt.

Die Blockentropien $H(A|A^n)$ fallen also exponentiell gegen h ab. Das bedeutet, daß man schon aus der Kenntnis weniger vorausgehender Symbole das nächste relativ gut vorhersagen kann und somit die regulären Sprachen auch in dieser informationstheoretischen Hinsicht eine geringe Komplexität aufweisen.

References

[1] Bialek, William, Ilya Nemenman und Naftali Tishby: *Predictability, complexity and learning.* Neural Computation, 13, 2001.

[2] Chomsky, Noam: *Three models for the description of language.* IRE Trans. on Information Theory, 2(3):113–124, 1956.

[3] Chomsky, Noam: *Aspects of the theory of syntax.* MIT Press, Cambridge, Mass., 1965.

[4] Cover, Thomas M. und Joy A. Thomas: *Elements of Information Theory.* Wiley, New York, 1991.

[5] Elman, Jeffrey L.: *Finding Structure in Time.* Cognitive Science, 14:179–211, 1990.

[6] Elman, Jeffrey L., Elizabeth A. Bates, Mark H. Johnson, Annette Karmiloff-Smith, Domenico Parisi und Kim Plunkett: *Rethinking Innateness: A Connectionist Perspective on Development.* MIT Press, Cambridge, Mass., 1996.

[7] Gold, E. Mark: *Language Identification in the Limit.* Information and Control, 10:447–474, 1967.

[8] Grassberger, Peter: *Toward a Quantitative Theory of Self-Generated Complexity.* International Journal of Theoretical Physics, 25(9):907–938, 1986.

[9] Hopcroft, John E. und Jerrey D. Ullmann: *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Mass., 1979.

[10] Wiles, Janet, Alan D. Blair und Mikael Bodén: *Representation Beyond Finite States: Alternatives to Push-Down Automata.* In Kolen, John F. und Stefan C. Kremer (Hrsgg.): *Dynamical Recurrent Networks.* IEEE Press, New York, 2001.