



Christina Klüver
Jürgen Klüver *Hrsg.*

Neue Algorithmen für praktische Probleme

Variationen zu Künstlicher Intelligenz und
Künstlichem Leben

 Springer Vieweg



Neue Algorithmen für praktische Probleme

Christina Klüver · Jürgen Klüver
(Hrsg.)

Neue Algorithmen für praktische Probleme

Variationen zu Künstlicher Intelligenz und
Künstlichem Leben

Hrsg.
Christina Klüver
CEO REBASK GmbH
Essen, Deutschland

Jürgen Klüver
Forschungsgruppe CoBASC
Essen, Deutschland

ISBN 978-3-658-32586-2 ISBN 978-3-658-32587-9 (eBook)
<https://doi.org/10.1007/978-3-658-32587-9>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2021

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Planung/Lektorat: Reinhard Dapper

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Vorwort

Das generelle Thema dieses Sammelbandes ist die Darstellung, wie drei von uns entwickelte neue Algorithmen aus den Bereichen „Künstliche Intelligenz“ und „Künstliches Leben“ auf praktische Probleme erfolgreich angewandt werden können. Es handelt sich dabei um ein sog. selbstorganisiert lernendes neuronales Netz (SEN), einen erweiterten evolutionären Algorithmus (RGA) und einen Algorithmus zur Ordnung von Daten (ANG).

Für die Entwicklung der Algorithmen waren theoretische Aspekte ausschlaggebend, um Lern- und/oder Adaptionprozesse stärker an dem anzupassen, wie Menschen lernen oder wie evolutionäre Prozesse nach neuen Erkenntnisse ablaufen. Insbesondere sollten diese Algorithmen „einfach“ sein, in dem Sinne, dass die Algorithmen ohne fundierte Mathematik- oder Informatikkenntnissen eingesetzt werden können.

Diese Algorithmen wären jedoch „graue Theorie“ geblieben, wenn im Laufe der Jahre nicht zahlreiche Studierende die Leidenschaft der Herausgeberin für diese Algorithmen geteilt hätten: Durch Diskussionen, Implementierungen, Tests, manchmal bis tief in die Nacht, sind diese Algorithmen lebendige Praxis geworden.

Durch weitere zahlreiche Modelle wurden auch die Algorithmen stets weiterentwickelt und ein Zwischenstand wird in diesem Sammelband gezeigt, gemäß dem Motto

Von Praktikern für Praktiker

Für uns als Konstrukteure der neuen Algorithmen ist es natürlich ungemein befriedigend zu sehen, wie erfolgreich diese zuerst etwas abstrakt wirkenden Innovationen in konkreten und praktisch relevanten Bereichen sein können. Nicht zuletzt deswegen möchten wir an dieser Stelle den verschiedenen Autoren/innen für ihr Engagement und ihre Mühe bei der Erstellung der Beiträge danken.

Es ist eine Ehre für uns, dass ein Verlag einen Sammelband herausgibt, indem die Beiträge die von uns entwickelten Algorithmen im Zentrum haben. Dafür möchten wir uns herzlich bei Herrn Reinhard Dapper, bei Herrn Rahul Ravindran, auch für seine Beharrlichkeit, und bei Herrn Pransanjit Das vom Springer Vieweg Verlag danken.

Wir widmen dieses Buch all den zahlreichen Studierenden und Praktikern, die uns bei der Entwicklung und bei dem Einsatz der Algorithmen geholfen haben.

Und nun:

Habeant sua fata libelli!

Essen
September 2020

Christina Klüver
Jürgen Klüver

Inhaltsverzeichnis

1	Einleitung: Thema mit Variationen	1
	Christina Klüver und Jürgen Klüver	
Teil I Einsatz des Self-Enforcing Networks (SEN)		
2	Teil I: KI – Das Self-Enforcing Network (SEN)	9
	Christina Klüver und Jürgen Klüver	
3	Bewertung und Auswahl von Vorgehensmodellen im IT-Projektmanagement – Ein Ansatz für die Unternehmenspraxis	21
	Christoph Albers	
4	Qualitätsverbesserung im Anforderungsmanagement durch Einsatz von Metriken	39
	Katrin Traue	
5	KI-gestützte Aufwandsschätzung in agilen IT-Projekten	55
	Matthias Köhler	
6	Ermittlung und Bewertung wesentlicher Aufwandstreiber für das Defect- Management – eine Fallstudie	77
	Guido Schwering	
7	Entscheidungsunterstützung bei Auswahlprozessen von Softwarekomponenten durch Self-Enforcing Networks (SEN)	97
	Kathrin Stein	
8	Einsatz eines Self-Enforcing Netzwerkes für die Ermittlung geeigneter Führungsstile auf Basis des „Process Communication“ Modells (PCM)	117
	Stefan Engels	

9	Erhöhung der Effizienz von agilen Teams unter Verwendung von Self-Enforcing Networks	135
	Christine Salzeller	
10	Entwicklung einer Konzeption zur Effektivitätsmessung von IT-Beratern	161
	Moritz Eifler	
11	Rekonstruktion der US-Wahlergebnisse 2016: Modellierung und Simulation der Prognosen	185
	Alexandar Schkolski, Mina Maria Zengin und Jan Demmer	
12	Meinungsprognosen mithilfe von sozialen Netzwerken – künstliche Intelligenz als neues Instrument zur Wahlprognose	207
	Erik Karger, Marko Kureljusic, Arda Cayci und Kevin Sigmund	
13	Entscheidungsunterstützungssystem zur Interpretation probabilistischer Wettervorhersagen für den Flughafen Frankfurt	229
	Dirk Zinkhan	
14	Logistische Regressionsanalysen und Self-Enforcing Networks zur Entdeckung von Akquisezielen in der deutschen Stahlindustrie durch Finanzkennzahlen	245
	Fatih Önder	
15	Analyse und Klassifikation von Voice Over IP-Angriffsdaten mit „ClustSEN“	265
	Waldemar Hartwig	
16	Datenanalyse von Arbeitszeiten aus Bilddateien mit Self-Enforcing Networks	281
	Daniel Büttner	
17	Bilderkennung von Verkehrszeichen mit Self-Enforcing Networks	305
	Björn Zurmaar	
18	Homogenitätsprüfung von LED-Lichtleitern durch Neuronale Netzwerke	325
	Sandra Thiemermann, Gregor Braun und Christina Klüver	
Teil II Einsatz des Regulatoralgorithmus (RGA)		
19	Künstliches Leben (KL) – Der Regulator Algorithmus (RGA)	343
	Christina Klüver und Jürgen Klüver	

20	Materialbedarfsplanung unter Berücksichtigung von Ressourcenkapazität und minimaler Losgröße durch einen RGA	351
	Matthias Hubert	
21	Variabilitätsmodellierung und Optimierung softwareintensiver Systeme durch einen Regulator Algorithmus (RGA)	373
	Ole Meyer	
22	Raumbelegungspläne mit einem Regulator Algorithmus	389
	Marcel Kleine-Boymann	
23	Webbasierte Raum- und Zeitplanung für schriftliche Prüfungen in der universitären Lehre	407
	Arne Hetzenegger und Firas Zaidan	
Teil III ANG und hybride Systeme		
24	Die Generierung von Datenordnungen durch den Algorithm for Neighborhood Generating (ANG)	431
	Christina Klüver und Jürgen Klüver	
25	Webbasierte Anwendung des Algorithm for Neighborhood Generating (ANG) zur Strukturierung und Analyse großer Datenmengen	441
	Jozsef Sütö und Christina Klüver	
26	Auswahl technischer Komponenten durch die Koppelung des „Algorithm for Neighborhood Generating“ (ANG) mit „Self-Enforcing Networks“ (SEN)	459
	Janis Höpken	
27	Epilog	481
	Christina Klüver und Jürgen Klüver	



Einleitung: Thema mit Variationen

1

Christina Klüver und Jürgen Klüver

Zusammenfassung

In diesem Sammelband geht es darum, neue Algorithmen aus den Bereichen der Künstlichen Intelligenz (KI) und des Künstlichen Lebens (KL) zu zeigen und deren praktische Anwendung zu zeigen; diese Algorithmen sind gewissermaßen Variationen zu etablierten KI- und KL-Algorithmen. Der wesentliche Aspekt des Bandes ist jedoch, dass in den Beiträgen exemplarisch gezeigt wird, dass und wie diese neuen Algorithmen auf praktische Probleme in sehr verschiedenen Bereichen erfolgreich eingesetzt werden können: Von der Modellierung sozialer Aspekte in der Softwareentwicklung bis zur Entscheidungsunterstützung, welche Start- und Landebahn an einem Flughafen ausgewählt werden soll; von der Analyse von Krankheitsverläufen bis zur Auswahl und Optimierung technischer Systeme, sowie Alternativen für die Bildbearbeitung. Praktische Probleme können durch diese Algorithmen in einer neuen Weise bearbeitet werden, wie anhand von 22 Beispielen demonstriert wird.

Schlüsselwörter

Künstliche Intelligenz (KI) · Maschinelles Lernen (ML) · Künstliches Leben (KL)

C. Klüver (✉)
REBASK GmbH, Forschungsgruppe COBASC, Essen, Deutschland
E-Mail: kluever@rebask.de

J. Klüver
Forschungsgruppe COBASC, Essen, Deutschland
E-Mail: juergen.kluever@uni-due.de

Die Begriffe der „Künstlichen Intelligenz“ (KI) und des „Maschinellen Lernens“ (ML) sowie in geringerem Maße des „Künstlichen Lebens“ (KL) haben in den letzten Jahrzehnten eine ungemeine Bekanntheit erreicht und sind weit über die einschlägigen wissenschaftlichen Expertengemeinschaften hinaus überaus populär geworden. Diese Entwicklungen haben natürlich – wie stets in derartigen Kontexten – zu einer Inflation in dem Sinne geführt, dass mittlerweile alle möglichen technischen Innovationen als „intelligent“ bezeichnet werden. Dies gilt insbesondere für Computertechniken, die es schon lange vor dieser Begriffsinflation gab und die jetzt als „intelligent“ bezeichnet werden. Das soll hier gar nicht weiter kommentiert werden.

Der hier vorliegende Sammelband hat im Kontrast zu vielen Publikationen mit hoch formulierten Zielen einen vergleichsweise bescheidenen Anspruch: Es geht darum, neue Algorithmen zu zeigen, die aus den Bereichen der KI und des KL stammen, und diese vor allem an praktischen realen Beispielen zu erläutern. Diese Algorithmen sind gewissermaßen Variationen zu etablierten KI- und KL-Algorithmen und sind für daran Interessierte möglicherweise schon deswegen relevant. Der wesentliche Aspekt dieses Bandes ist jedoch, dass in den Beiträgen exemplarisch gezeigt wird, dass und wie diese neuen Algorithmen auf praktische Probleme in sehr verschiedenen Bereichen erfolgreich eingesetzt werden können. Insbesondere wird ersichtlich, dass eine praktische Verwendung dieser Algorithmen eigene grundlegende Kenntnisse in Informatik oder entsprechenden Bereichen nicht voraussetzt. Die Beiträge dieses Bandes stammen überwiegend nicht von „reinen“ Informatikern, sondern vor allem von Absolventen wirtschaftswissenschaftlicher Disziplinen (einschließlich der Wirtschaftsinformatik).

Beginnen wir mit einer Begriffsklärung, die insbesondere angesichts der häufig schon als „KI-Hype“ bezeichneten Vielfalt des Begriffs „intelligent“ sinnvoll und notwendig erscheint. Dazu kommt noch, dass „Maschinelles Lernen“ (ML) in diesem Kontext ebenfalls verwendet wird – häufig als synonym mit Künstlicher Intelligenz und ebenfalls als eigentlicher Fundamentalbegriff. Wir orientieren uns in diesem Band an einer begrifflichen Unterscheidung, die im VDI-Statusreport 2019 definiert worden ist (VDI 2019): Als KI werden alle Ansätze bezeichnet, die sich *heuristisch* am menschlichen Denken mit entsprechenden algorithmischen Techniken orientieren und die vor allem zur Basis stets ein *Modell* des jeweiligen Gegenstandsbereichs haben.

Wenn z. B. für eine Offshore Windkraftanlage ein günstiger Standort mithilfe eines neuronalen Netzes bestimmt werden soll – ein Beispiel, auf das wir noch zurückkommen werden –, dann müssen in einem Modell des Problembereichs insbesondere die wesentlichen Parameter wie Tiefe des Wassers, Entfernung von der Küste etc. in einem Modell bestimmt werden, um ein entsprechendes neuronales Netz einsetzen zu können. Jedoch auch weitere Entscheidungskriterien wie Sicherheit, politische Lage etc., die nicht nur objektiv, wie die Messung der Tiefe, angegeben werden können. Die Konstruktion der entsprechenden Modelle bedarf häufig theoretischer Überlegungen, um die Validität des Modells zu garantieren.

Im Maschinellen Lernen dagegen, bei dem häufig gleiche Techniken verwendet werden wie bei KI-Ansätzen, geht es „nur“ um die Leistungsfähigkeit der jeweiligen

Techniken bei der Analyse großer Datenmengen; das Hauptproblem besteht hier in der numerischen Bestimmung der wesentlichen Parameter. Man kann, wenn man dieser Unterscheidung folgt, KI-Ansätze in der Orientierung an kognitivistischen Lerntheorien verstehen, ML-Ansätze dagegen in der Tradition des Behaviorismus, also nur an der Performanz der künstlichen Systeme. Es sei hier bereits ausdrücklich hervorgehoben, dass die Beiträge in diesem Band ausschließlich der hier beschriebenen KI-Orientierung folgen.

Im Bereich des Künstlichen Lebens gibt es diese Unterscheidung nicht und zwar aus dem einfachen Grund, dass zumindest in der deutschsprachigen Tradition zwischen KI und KL häufig nicht explizit unterschieden wird. Zwar gab es vor allem in den USA in den achtziger und neunziger Jahren im Rahmen der Komplexitätsforschung verschiedene Studien zum „Artificial Life“ (z. B. Langton 1994), aber im Gegensatz zur „Artificial Intelligence“ blieb dieser Bereich eher sekundär. Wir verweisen in diesem Band deswegen auf die Tradition des KL mit Varianten zu den dort entwickelten Algorithmen, weil wir auch damit der inflationären Verwendung des Intelligenzbegriffs etwas Präzision entgegensetzen wollen: Evolutionäre Algorithmen, die wir hier auch demonstrieren werden, haben nun einmal mit Intelligenz wenig bis gar nichts zu tun, aber mit Grundprinzipien der biologischen Evolution sehr viel.

Der Vollständigkeit halber sei hier noch darauf hingewiesen, dass im Bereich der KI zusätzlich unterschieden wird zwischen der sog. *symbolischen KI* und der *sub-symbolischen*. Als symbolische KI ist vor allem der Ansatz zu verstehen, der mit sog. Expertensystemen oder auch regelbasierten Systemen arbeitet, bei denen es um die Darstellung von und Operationen mit logischen Strukturen geht; die Einheiten dieser Systeme sind häufig symbolische Begriffe – daher der Name. Neuronale Netze dagegen sind subsymbolische Systeme in dem Sinne, dass ihre Einheiten, die sog. Neuronen, gewöhnlich nicht eigenständige Symbole sind, sondern erst durch ihre Verknüpfung mit anderen Neuronen so etwas wie Sinnhaftigkeit erreichen. Wir haben in diesem Band unter dem Stichwort KI ausschließlich Beispiele, wie ein bestimmter von uns entwickelter Netzwerktypus praktisch eingesetzt werden kann; es handelt sich also um einen subsymbolischen Ansatz.

Es ist sinnvoll, in der KI-Forschung und Entwicklung zwischen verschiedenen Zielen zu unterscheiden und zwar allgemeine Ziele, die die Logik des jeweiligen Problems festlegen:

- a) Gegeben sind bei diesem Ansatz extern definierte Ziele, an denen sich der Erfolg der jeweiligen Systeme ausrichtet. Gemeint ist damit, dass durch Variation bestimmter Systemparameter sich die Performanz des Systems ändert und zwar derart, dass der Output des Systems möglichst nahe an die externen Ziele herankommt. Dies wird im erfolgreichen Fall erreicht durch bestimmte *Lernregeln*, die die interne Struktur solange ändern, bis der Output des Systems der Zielvorgabe möglichst nahe kommt. Bekannt ist bei diesem Ansatz die Bezeichnung *überwachtes Lernen*, also eine Systemvariation, die wie im Fall z. B. des schulischen Lernens ständig extern kontrolliert wird.

Da es hier im Wesentlichen um die erfolgreiche Performanz des jeweiligen Systems geht, spielt dieser Ansatz vor allem beim Maschinellen Lernen eine wesentliche Rolle. Die hier meistens verwendete Lernregel ist die sog. *Backpropagation Regel* und deren zahlreiche Variationen, die häufig als *das* Paradigma erfolgreicher Lernregeln angesehen werden. Man kann sich das methodische Vorgehen bei diesem Ansatz an dem wichtigen Anwendungsbereichs des autonomen Fahrens verdeutlichen: Es werden klassische Verkehrssituationen vorgegeben und das Steuerungssystem lernt durch Verwendung einer Backpropagation Variante, sich in dieser Situation bzw. einer ähnlichen möglichst adäquat zu verhalten.

Mit einer Ausnahme orientieren sich die Beiträge in diesem Band nicht an diesem methodischen Ansatz, sondern an einem der beiden folgenden.

- b) Bei diesem Ansatz handelt es sich um die *Optimierung* eines bestimmten Systemverhaltens, wobei die Kriterien für die Optimierungsprozesse wieder extern vorgegeben sind. In Analogie zum Begriff des überwachten Lernens wird hier auch häufig der Begriff *verstärkendes Lernen* verwendet. Gemeint ist damit, dass bestimmte Systemparameter oder auch Struktureigenschaften des entsprechenden Systems systematisch verstärkt werden – bzw. abgeschwächt –, bis ein Optimum erreicht worden ist, das nicht mehr verbessert werden kann. Man unterscheidet hier gewöhnlich zwischen *lokalen* und *globalen Optima*: Während lokale Optima grundsätzlich noch durch entsprechende Variationen der verwendeten Algorithmen verbesserungsfähig sind, sofern man dies wünscht, sind globale Optima das Ende der Optimierungsprozesse – die Systeme haben in einem abstrakten Optimierungsraum einen *Attraktor* erreicht. Wesentlich ist hier, dass im Gegensatz zum überwachten Lernen ein explizites Ziel nicht vorgegeben ist, da es häufig auch gar nicht bekannt ist. Vorgegeben sind die Kriterien für „besser oder schlechter“.

Es gibt natürlich schon seit langem zahlreiche Optimierungsverfahren. Für die Thematik dieses Bandes sind insbesondere aus dem Bereich Künstliches Leben die sog. *Evolutionären Algorithmen* wichtig (vgl. z. B. Klüver et al. 2012), von denen der bekannteste der *Genetische Algorithmus (GA)* ist, entwickelt in den sechziger Jahren von John Holland. Die evolutionären Algorithmen sind methodisch am Prinzip der biologischen Evolution orientiert – daher der Name: Vorgegeben sind a) die sog. Fitnesskriterien, also Maße für die Optimierungsziele, und b) „Populationen von Individuen“. Die „Individuen“ repräsentieren die zu optimierenden Systeme; die Individuen einer Population werden den „genetischen Operationen“ der Mutation und den Kreuzungen unterzogen. Dies geschieht so lange, bis ein Optimierungsziel erreicht wird bzw. bis der Optimierungsprozess in einem Attraktor angelangt ist.

Wir stellen in diesem Band in Teil II einen neuartigen evolutionären Algorithmus vor, sodass diese allgemeinen Hinweise genügen können.

- c) Bei den beiden geschilderten Ansätzen geht es jeweils um die Variation eines bestimmten Systems aufgrund externer Kriterien – seien dies explizit vorgegebene Ziele wie bei (a) oder generelle Optimierungskriterien wie bei (b). Dies ist beim Typus (c) etwas anders, da es hier nicht primär um Systemveränderungen

geht, sondern vor allem um die *Ordnung von Daten*. Allerdings geht es hier auch um Variationen des Systems, jedoch primär auf der Basis selbstorganisierender Prozesse. Dieser Typus wird gewöhnlich als *selbstorganisiertes Lernen* bezeichnet, da die Operationen derartiger Systeme nicht extern vorgegebenen Zielen folgen. Der bekannteste Algorithmus ist die „Kohonen-Karte“, auch als Self Organized Map (SOM) bezeichnet, von Taavo Kohonen.¹

Die allgemeine Grundlogik dieser „selbstorganisierten“ Netzwerke lässt sich folgendermaßen beschreiben: Gegeben ist eine Menge von Daten, die als *Objekte mit Attributen* dargestellt werden. Die Attribute, also Eigenschaften der Objekte, charakterisieren die Ähnlichkeiten der Objekte bzw. auch deren Unterschiedlichkeit. Die Aufgabe der selbstorganisiert operierenden Netze ist nun, die Objekte gemäß den in den Attributen enthaltenen Ähnlichkeiten der Objekte zusammen zu fassen – ähnliche Objekte beieinander und unähnliche Objekte voneinander entfernt. Man kann dies mathematisch charakterisieren, dass die Objekte in einem abstrakten topologischen Raum situiert sind und dass durch die Netzwerkoperationen eine Clusterung der Objekte gemäß ihren Ähnlichkeiten vorgenommen wird. Dies wird noch am Beispiel der SEN Operationen im nächsten Teil verdeutlicht.

„Ähnlichkeit“ und „Unähnlichkeit“ können natürlich sehr unterschiedliche Aspekte einer Datenordnung erfassen. Auch das wird anhand des selbstorganisiert operierenden SEN verdeutlicht. Entsprechend kann der Begriff „Ordnung von Datenmengen“ inhaltlich sehr verschiedene Bedeutungen haben – man kann eine Menge von Tieren nach Größe, biologischen Eigenschaften wie Raubtiere etc. ordnen. Das wird an den Beiträgen in Teil I sehr deutlich.

Wenn bestimmte Attribute für die Klassifizierung der entsprechenden Objekte besonders wichtig sind, kann man diese zusätzlich mit einem *cue validity factor* charakterisieren, also einem numerischen Wert, der die besondere Bedeutung dieses Attributs hervorhebt. Wenn z. B. Katzen von Hunden in einer Klassifizierung unterschieden werden sollen, ist es sinnvoll, die jeweiligen Geräusche – Bellen versus Miauen – hervorzuheben. Diesen Begriff kann man etwas schwerfällig als „Schwerpunktwert“ übersetzen; wir bezeichnen ihn mit dem englischen Original, der von der Kognitionspsychologin Rosch stammt (Rosch 1973). Das wird in Teil I noch im Detail erläutert.

In den meisten Beiträgen dieses Bandes wird gezeigt, wie inhaltlich äußerst verschiedene Probleme aus unterschiedlichen Praxisbereichen durch den Einsatz des von uns entwickelten *Self-Enforcing Networks (SEN)* erfolgreich bearbeitet worden sind.

¹Das Prinzip des selbstorganisierten Lernens ist, neben dem des überwachten Lernens, auch Teil der ART-Netze (Adaptive Resonance Theory) von Stephen Grossberg. In Kap. 18 wird eine Variation der Koppelung zwischen überwacht- und selbstorganisiert lernenden Netzwerken gezeigt werden.

Wir haben für die Ordnung von Datenmengen allerdings noch einen weiteren Algorithmus konstruiert, der in diesem Band auch dargestellt wird, nämlich den von uns sogenannten *Algorithm for Neighborhood Generating (ANG)*. Dieser Algorithmus ist keine neue Variation herkömmlicher Netzwerktypen, sondern lässt sich als eine Orientierung an der Logik von *Zellularautomaten* verstehen. Dies wird in Teil III wieder an Anwendungsbeispielen dargestellt.

Die in den verschiedenen Beiträgen dargestellten Anwendungen der einzelnen Algorithmen sind zu einem großen Teil der beruflichen Praxis der Autoren entnommen. Die Autorinnen und Autoren sind ausnahmslos ehemalige Studierende von uns – sonst wären sie ja auch mit den neuen Algorithmen nicht so vertraut, wie sie es für die Arbeiten brauchten. Indem es sich bei den Beiträgen ausschließlich um Realprobleme handelt, wird auch gleichzeitig demonstriert, wie gut für unterschiedliche Praxisfelder diese immer noch relativ neuen Algorithmen geeignet sind.

Der Band ist in drei Teile aufgeteilt: Teil I gehört zum Themenbereich der KI und zeigt die Anwendungsmöglichkeiten des erwähnten Self-Enforcing Networks bei der Ordnung von Datenmengen. Teil II zeigt die Optimierungsfähigkeiten unseres Regulator Algorithmus aus dem Bereich des KL und in Teil III schließlich geht es wieder um Ordnungsprobleme mit ANG, einem topologischen Algorithmus wieder aus dem Bereich KL. In dem letzten Beitrag wird gezeigt, wie die Koppelung von SEN und ANG nicht nur gelingt, sondern ebenso fruchtbar eingesetzt werden kann.

Beginnen wir also mit den eigentlichen Inhalten dieses Bandes, denn:

„Grau teurer Freund ist alle Theorie
Doch grün des Lebens goldener Baum“

Goethe, Faust I.

Literatur

- Klüver C, Klüver J, Schmidt J (2012) Modellierung komplexer Prozesse durch naturanaloge Verfahren. *Soft Computing und verwandte Techniken*, 2. Aufl. Springer Vieweg, Wiesbaden
- Langton C (Hrsg) (1994) *Artificial Life III*. Addison Wesley, Reading
- Rosch E (1973) Natural categories. *Cogn Psychol* 4:328–350
- VDI Statusreport (2019) Maschinelles Lernen. Künstliche Intelligenz mit neuronalen Netzen in optischen Mess- und Prüfungssystemen. VDI Statusreport November 2019. VDI / VDE-H Gesellschaft Mess- und Automatisierungstechnik. <https://www.vdi.de/ueber-uns/presse/publikationen/details/kuenstliche-intelligenz-mit-neuronalen-netzen-in-optischen-mess-und-pruefungssystemen>

Teil I

Einsatz des Self-Enforcing Networks (SEN)



Teil I: KI – Das Self-Enforcing Network (SEN)

2

Christina Klüver und Jürgen Klüver

Zusammenfassung

In dieser Einleitung werden die formalen Grundlagen von Self-Enforcing Networks (SEN) gezeigt, auf denen die inhaltlichen Beiträge aufbauen.

Schlüsselwörter

Self-Enforcing Networks · selbstorganisiertes Lernen

2.1 Einleitung

Die Motivation, dies neue selbstorganisierte Netzwerk zu entwickeln, entstand vor allem aus theoretischen Gründen. Künstliche neuronale Netzwerke sind meistens sehr komplex. Durch die zufällige Generierung der sogenannten Gewichtsmatrix, in der das gesamte „Wissen“ eines Netzwerkes enthalten ist, und anschließenden Veränderung dieser Gewichtsmatrix durch die jeweilige Lernregel, ist es schwer nachzuvollziehen, wie die Netzwerke operieren. Die Nachvollziehbarkeit und Interpretation der Ergebnisse sind daher für Laien auf dem Gebiet meistens nicht gegeben.

Da wir daran interessiert sind, völlig unterschiedliche Probleme zu modellieren, insbesondere auch kognitive Prozesse, war es uns wichtig, ein Netzwerk zu entwickeln, das

C. Klüver (✉)
REBASK GmbH, Forschungsgruppe COBASC, Essen, Deutschland
E-Mail: kluever@rebask.de

J. Klüver
Forschungsgruppe COBASC, Essen, Deutschland
E-Mail: juergen.kluever@uni-due.de

nach bekannten *kognitiven* Lernregeln sich selbstorganisiert das Wissen aneignet und klassifiziert.

Das Ergebnis ist das Self-Enforcing Network – SEN.

Die Annahme des SEN von Studierenden für die Entwicklung von Modellen und Analyse der Daten in ganz unterschiedlichen Disziplinen, sowie der Einsatz für die Lösung praktischer Probleme in diversen Unternehmen, bestätigt uns, einen Algorithmus entwickelt zu haben, der ohne Vorerfahrung oder Programmierkenntnisse intuitiv verwendet werden kann. Dies ist insbesondere auch dank der Implementierung des Algorithmus von Björn Zurmaar möglich, der es Anwendern erlaubt, sich schnell mit der „SEN-Software“ vertraut zu machen.

Anhand dieser SEN-Software werden die Konzepte des Netzwerkes im Folgenden erläutert.

2.2 Bestandteile des SEN und der SEN-Software

In der Einleitung wird „SEN“ als Bezeichnung des von uns entwickelten Algorithmus verwendet. Im Folgenden werden die Bestandteile des SEN anhand der SEN-Software aus Anwendersicht erläutert. In den Beiträgen beziehen sich die Autoren auf die SEN-Software und deren Einsatz für die entwickelten Modelle bzw. für die Analyse der Daten; „SEN“ wird zugleich synonym für das Netzwerk verwendet.

2.2.1 Die semantische Matrix

Bei den Daten, für deren Ordnung SEN eingesetzt wird, handelt es sich, wie unter (c) in der allgemeinen Einleitung bereits angemerkt, um Objekte, die durch bestimmte Attribute charakterisiert sind. Die Grundlage einer SEN-Konstruktion ist entsprechend eine *semantische Matrix*, deren Zeilen die Objekte und die Spalten die entsprechenden Attribute repräsentieren. Die Werte in der Matrix drücken den *Zugehörigkeitsgrad* eines Attributs zum jeweiligen Objekt aus.

Illustriert wird dies anhand eines sehr kleinen „Modells“ für die Beiträge in diesem Sammelband: Als Attribute werden die eingesetzten Algorithmen (Self-Enforcing Networks (SEN), Regulator Algorithmus (RGA), Algorithm for Neighborhood Generating (ANG)), sowie deren Einsatz (Modellierung, Optimierung, Analyse großer Datenmengen und Bildbearbeitung) angegeben. In der semantischen Matrix werden die Beiträge als Objekte definiert und deren Anzahl sowie deren Einsatz numerisch angegeben (Abb. 2.1).

Aus der semantischen Matrix (Abb. 2.1 rechts) ist erkennbar, dass in 16 Beiträgen SEN eingesetzt wird, in denen 13mal ein Modell entwickelt und präsentiert wird, in einem Beitrag große Datenmengen analysiert werden und in zwei Beiträgen der Einsatz

The screenshot shows two windows from the SEN software. The left window, titled 'Attribute', displays a table of attributes with their standard, minimum, and maximum values, and their coding ranges. The right window, titled 'Semantische Matrix', shows a table of normalized data for three objects (SEN, RGA, ANG) across seven attributes.

Name	Standard	Minimum	Maximum	Kodierung
SEN	0,00	0,00	16,00	[-1; 1]
RGA	0,00	0,00	4,00	[-1; 1]
ANG	0,00	0,00	2,00	[-1; 1]
Modellierung	0,00	0,00	15,00	[-1; 1]
Optimierung	0,00	0,00	4,00	[-1; 1]
Analyse großer Datenmengen	0,00	0,00	3,00	[-1; 1]
Bildbearbeitung	0,00	0,00	2,00	[-1; 1]

Objekt Name	SEN	RGA	ANG	Modellierung	Optimierung	Analyse großer Datenmengen	Bildbearbeitung
Beitrag SEN	16,00	0,00	0,00	13,00	0,00	1,00	2,00
Beitrag RGA	0,00	4,00	0,00	0,00	4,00	0,00	0,00
Beitrag ANG	1,00	0,00	2,00	0,00	0,00	2,00	0,00

Abb. 2.1 Attribute, Objekte und Semantische Matrix des SEN

des SEN für die Bildbearbeitung vorgestellt wird. Der RGA steht in 4 Beiträgen für Optimierungsprobleme im Mittelpunkt und ANG wird 2mal für die Analyse großer Datenmengen sowie einmal in der Koppelung mit SEN eingesetzt.

Dieses sehr einfache Modell des Methodeneinsatzes kann natürlich verfeinert werden, indem mehr Einzelheiten aufgenommen werden und die semantische Matrix wesentlich differenzierter ausgefüllt wird.

Da es sich in diesem Fall um ein sehr kleines Modell handelt, das direkt in der SEN-Software entwickelt werden kann, müssen die Min-Max-Werte für die Attribute (Abb. 2.1 links) manuell eingestellt werden; bei einem Import der Modelle bzw. der Daten werden diese automatisch ausgelesen und eingetragen.

Normalisierung der Daten Die Daten in der semantischen Matrix werden automatisch normalisiert. Der Wertebereich kann für die Attribute individuell angegeben werden; in den meisten Fällen wird dieser jedoch binär im Intervall zwischen [0,1] oder bipolar im Intervall [-1,1] kodiert, wie es in Abb. 2.1 der Fall ist. Dies geschieht gemäß der Formel

$$v_{norm} = \frac{v_{raw} - r_{min}}{r_{max} - r_{min}} * (n_{max} - n_{min}) - n_{min} \quad (Gl. 2.1)$$

Eine bi-polare Kodierung bietet sich insbesondere an, wenn die Wertebereiche sehr stark abweichen. Im Normalfall werden die Objekte am stärksten aktiviert, deren Verbindungen zu den Attributen hohe Werte aufweisen; dies wird durch die Normalisierung im Wertebereich zwischen -1 und 1 verhindert.

Nach der Konstruktion der semantischen Matrix werden weitere Funktionen und Parameter für den Lernprozess bestimmt.

2.2.2 Externe Topologie und Dynamik des SEN

Mit Topologie ist die sog. „externe Topologie“ gemeint und damit die Verteilung der Neuronen (Attribute und Objekte) auf die Schichten (Eingabe- und Ausgabeschicht) sowie deren „Informationsfluss“.

In den meisten Fällen kann ein SEN als zweidimensionales Netzwerk verstanden werden, das je nach Verbindungen als vorwärtsgerichtet (feed-forward), rückwertegerichtet (feed-back) oder auch rekurrent charakterisiert ist. In diesen Fällen bilden die Attribute die Eingabeschicht (Inputschicht) und die Objekte entsprechend die Ausgabeschicht (Outputschicht).

Die Dynamik des Netzes wird allgemein durch dessen Topologie bestimmt. Sind z. B. nur die Attribute mit den Objekten verbunden, dann ist die Dynamik des Netzwerks eine reine feed-forward Dynamik. Bei zusätzlichen Verbindungen von den Objekten zu den Attributen handelt es sich um eine feed-back Dynamik und bei Verbindungen der Attribute oder auch der Objekte zueinander um eine rekurrente Dynamik.

In allen Anwendungsfällen in diesem Band ist eine feed-forward Dynamik hinreichend sowie ein zweidimensionales Netzwerk (auch als zweischichtiges Netz bezeichnet). In Abb. 2.1 sind demnach die Attribute Bestandteil der Eingabeschicht und die Objekte sind der Ausgabeschicht zugeordnet; darüber hinaus ist nur zwischen den Objekten und den Attributen ein numerischer Wert angegeben.

Ein Vektor in der semantischen Matrix entspricht demnach einem Objekt mit den dazugehörigen Attribute und deren Ausprägung.

2.2.3 Funktionen

Neben der Topologie wird die Dynamik des Netzes durch eine *Aktivierungsfunktion* festgelegt. Es gibt grundsätzlich verschiedene Funktionen¹, die in einem neuronalen Netzwerk eingesetzt werden, unabhängig vom Netztypus (Kap. 18). Die Aktivierungsfunktion hat einen entscheidenden Einfluss auch auf den Lernprozess, daher wird diese Funktion als besonders wichtig erachtet. Diese ist auch entscheidend für die *Intensität* der Aktivierung einzelner Neuronen (in diesem Fall der Objekte).

Für eine SEN Konstruktion stehen mehrere unterschiedliche Funktionen zur Verfügung; in allen hier aufgeführten Fällen bedeuten a_j den Aktivierungswert des empfangenden Neurons j , a_i den Aktivierungswert des sendenden Neurons i und w_{ij} wie gewöhnlich bei neuronalen Netzwerken den Gewichtswert (weight) der Verbindung zwischen i und j .

Die Funktionen a), b) und c) sind Standardfunktionen in der Neuroinformatik, die Funktionen d), e) und f) sind von uns zusätzlich entwickelt worden:

a) Lineare Funktion

$$a_j = \sum w_{ij} * a_i \quad (\text{Gl. 2.2})$$

¹Zuerst wird immer der Nettoinput für ein Neuron berechnet (Netto-Inputfunktion), zweitens wird eine Aktivierungsfunktion und drittens eine Ausgabefunktion eingesetzt.

b) Tangens hyperbolicus

$$a_j = \tanh(\text{net}_j) = \frac{e^{(\text{net}_j)} - e^{(-\text{net}_j)}}{e^{(\text{net}_j)} + e^{(-\text{net}_j)}} \quad (\text{GI. 2.3})$$

c) Logistische Funktion

$$a_j = \frac{1}{1 + e^{-\text{net}_j}} \quad (\text{GI. 2.4})$$

d) Lineare Mittelwertfunktion (LMF)

$$a_j = \sum \frac{w_{ij} * a_i}{k}, k = \text{Anzahl der Verbindungen} \quad (\text{GI. 2.5})$$

e) Logarithmisch-lineare Funktion

$$a_j = \sum \begin{cases} \lg_3(a_i + 1) * w_{ij}, & \text{wenn } a_j \geq 0 \\ \lg_3(|a_i + 1|) * -w_{ij}, & \text{sonst} \end{cases} \quad (\text{GI. 2.6})$$

Der Logarithmus ist als „dämpfender Effekt“ eingeführt worden; die Basis 3 dient dazu, dass die Werte weder zu groß noch zu klein werden.

f) Verstärkende Aktivierungsfunktion (Enforcing Activation Function – EAF)

$$a_j = \sum_{i=1}^n \frac{w_{ij} * a_i}{1 + |w_{ij} * a_i|} \quad (\text{GI. 2.7})$$

In Kap. 17 wird eine zusätzliche Funktion von Björn Zurmaar eingeführt, nämlich die „Relative Funktion“, die sich zum Beispiel für die Bildbearbeitung sehr gut eignet.

Die Wahl einer geeigneten Aktivierungsfunktion ist problemabhängig. In den meisten Beiträgen in diesem Sammelband wird die logarithmisch-lineare Funktion eingesetzt, da diese die Wertebereiche der Endaktivierung nicht zu stark anwachsen lässt (wie die lineare Funktion) und zugleich eine differenzierte Aktivierung der Objekte zulässt. Ein Gegensatz dazu ist zum Beispiel die lineare Mittelwertfunktion, die praktisch die Werte aggregiert.

2.2.4 Der Lernprozess mit der Self-Enforcing Rule

Die Operationen eines SEN starten damit, dass die Werte aus der semantischen Matrix in die Gewichtswerte der SEN Gewichtsmatrix transformiert werden. Falls ein Objekt O das Attribut A nicht hat, dann ist der Wert der semantischen Matrix $v_{sm} = 0$ und der entsprechende Gewichtswert $w_{oa} = 0$. Dieser Wert bleibt konstant. In allen anderen Fällen ist der Gewichtswert

$$w_{oa} = c * v_{sm} \quad (\text{GI. 2.8})$$

c ist eine Konstante, die vom Benutzer als Parameter eingestellt werden kann. Sie entspricht der bekannten „Lernrate“ aus der Neuroinformatik und kann festgelegt werden im Intervall $0 < c < 1$.

Die Standardlernregel, die entsprechend dem jeweiligen Problem die Werte der Gewichtsmatrix variiert, ist

$$w(t+1) = w(t) + \Delta w \text{ und} \quad (\text{Gl. 2.9})$$

$$\Delta w = c * v_{sm}$$

Falls $w(t)=0$, dann ist $w(t+1)=0$ für alle weiteren Lernschritte.

Als Lernrate genügt in den meisten Fällen $c=0,1$; problemabhängig wird diese entweder manuell oder automatisiert eingestellt.

Wenn man die Bedeutung eines oder mehrerer Attribute besonders hervorheben will, kann ein bereits in der allgemeinen Einleitung erwähnter cue validity factor (cvf) eingefügt werden. Die Gl. 2.8 wird dann

$$\Delta w = c * w_{oa} * cvf_a \quad (\text{Gl. 2.10})$$

Die Gewichtsmatrix für das kleine Netzwerk (mit dem $cvf=1$ für alle Attribute) aus Abb. 2.1 wird in Abb. 2.2 dargestellt.

In Abb. 2.2 wird eine sogenannte „kompakte“ Ansicht dargestellt, da diese für die meisten Anwender ausreichend ist. In der Expertenansicht wird die vollständige Matrix gezeigt, in der zusätzlich die Topologie des Netzwerkes verändert werden kann.

SER steht für die Lernregel und diese „Selbstverstärkung“ (Self-Enforcing Rule) bedeutet einerseits, dass das eigentliche Netzwerk durch die Transformation der semantischen Matrix in die Gewichtsmatrix generiert wird (die 0-Werte zu Beginn in der Gewichtsmatrix werden durch die Lernregel selbst verändert); andererseits verstärkt sich das Netzwerk selbst durch die Anzahl der Lernschritte bzw. durch die Lernrate.

► SEN-Charakteristikum

Die Besonderheit des SEN besteht darin, dass die Gewichtsmatrix *nicht*, wie bei anderen Netzwerktypen üblich, per Zufall generiert wird. Das „Wissen“ für das Netzwerk wird in der semantischen Matrix eingegeben; das SEN transformiert dieses Wissen über die Lernregel 1:1 in das eigentliche Netzwerk. Dadurch ist es jederzeit möglich, den Lernprozess und die Ergebnisse des SEN zu rekonstruieren.



	SEN	RGA	ANG	Modellierung	Optimierung	Analyse großer Datenmengen	Bildbearbeitung
Beiträge SEN	0,10	-0,10	-0,10	0,07	-0,10	-0,03	0,10
Beiträge RGA	-0,10	0,10	-0,10	-0,10	0,10	-0,10	-0,10
Beiträge ANG	-0,09	-0,10	0,10	-0,10	-0,10	0,03	-0,10

Abb. 2.2 Gewichtsmatrix

Nach dem Lernprozess können neue Vektoren als „Eingabevektoren“ (Input) definiert werden. Diese sollen durch SEN gemäß ihrer Ähnlichkeit zu den gelernten Objekten (Vektoren der semantischen Matrix) klassifiziert werden.

Zusammenfassung Die Operationen eines SEN Modells bestehen demnach in a) der Konstruktion der problemspezifischen semantischen Matrix, b) der Transformation der Werte der semantischen Matrix gemäß der Gl. 2.8 und c) der Lernschritte gemäß Gl. 2.9 bzw. Gl. 2.10. Das Ergebnis sind die Endaktivierungswerte der Neuronen, die die Objekte repräsentieren.

Die Einfügung eines neuen Objekts wird bestimmt durch den Vergleich der Attributwerte des neuen Objekts mit den Endaktivierungswerten der bereits vorhandenen Objekte.

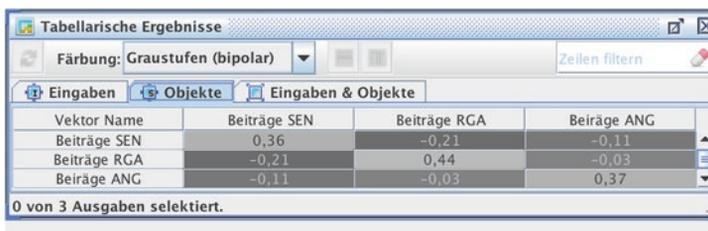
2.2.5 Visualisierungsmöglichkeiten

Zur raschen Übersicht der Ergebnisse eines SEN stehen mehrere Visualisierungsmöglichkeiten zur Verfügung; die technischen Details werden hier nicht weiter ausgeführt.

a) *Tabellarische Übersicht* Eine tabellarische Übersicht, bei der die Endaktivierungswerte der Objekte angezeigt werden; je größer die Endaktivierungen sind, desto relevanter sind die Objekte.

Die Ergebnisse des kleinen Beispiels sehen wie folgt aus (Abb. 2.3).

b) *Kartenvisualisierung* Eine „Kartenvisualisierung“, bei der die Objekte nach Ähnlichkeiten geordnet sind (konstruiert von Björn Zurmaar) und eine „topografische“ Gesamtübersicht ermöglichen.² Die Berechnung der Abstände zwischen den Objekten erfolgt nach der Euklidischen Distanz (Abb. 2.4).



Vektor Name	Beiträge SEN	Beiträge RGA	Beiträge ANG
Beiträge SEN	0,36	-0,21	-0,11
Beiträge RGA	-0,21	0,44	-0,03
Beiträge ANG	-0,11	-0,03	0,37

0 von 3 Ausgaben selektiert.

Abb. 2.3 Tabellarische Übersicht

²Dies wird detaillierter dargestellt in Klüver 2016.



Abb. 2.4 Kartenvisualisierung

Die drei Objekte (Algorithmen) sind relativ gleich voneinander entfernt visualisiert.

Die nächsten Visualisierungsoptionen werden nur dann aktiviert, wenn dem SEN ein neuer Eingabevektor präsentiert wird.

Beispiel: Ein Eingabevektor enthält alle Attribute der semantischen Matrix, die mit 0 belegt sind. In der Anwendung werden die Werte entweder automatisiert importiert oder manuell eingegeben, wie in Abb. 2.5 dargestellt.

In diesem Fall wird nur bei der Bildbearbeitung ein Wert eingegeben und als Ergebnis wird erwartet, in welchen Beiträgen es um Bildbearbeitung geht.

c) Eine Visualisierung im „Zentrummodus“: Hier geht es vor allem darum, neue Objekte in das System einzufügen. Die übersichtlichste Methode besteht darin, den Vektor des oder der neuen Objekte in das Zentrum des Gesamtbildes zu platzieren, während die bereits eingefügten Objekte an der Peripherie eingeordnet werden. Anschließend werden die geometrischen Distanzen der bereits vorhandenen Vektoren zum Zentrum bestimmt und in der Visualisierung gezeigt, d. h. die entsprechenden Objekte werden zum Zentrum gezogen.

Diese Visualisierung haben wir als „Input Centered Modus“ bezeichnet (Klüver et al. 2012, S. 152).

Man kann dies auch so verstehen, dass semantische Beziehungen zwischen den Objekten in geometrische übersetzt werden. Wir werden am Ende dieser Einleitung ein reales Beispiel dafür zeigen, in Abb. 2.6 ist die Visualisierung des kleinen Beispiels.

Im Zentrum befindet sich der Eingabevektor und die 3 Objekte werden unterschiedlich stark zum Zentrum angezogen. Die Aktivierung der Objekte ist aufgrund der Parametereinstellungen nicht sehr hoch und daher ist die Visualisierung auf Anziehung nicht



Sele...	Vektor Name	SEN	RGA	ANG	Modellierung	Optimierung	Analyse großer Datenmengen	Bildbearbeitung
<input checked="" type="checkbox"/>	Algorithmus für Bildbearbeitung	0,00	0,00	0,00	0,00	0,00	0,00	2,00

1 von 1 Eingaben selektiert.

Abb. 2.5 Neuer Eingabevektor



Abb. 2.6 SEN-Visualisierung

eindeutig. Bei genauer Betrachtung wird jedoch das Objekt „Beiträge SEN“ am stärksten vom Zentrum angezogen.

Die Visualisierungen dienen einer schnellen Übersicht; eine genaue Berechnung der Aktivierungswerte wird durch die Ranking- sowie durch die Distanzen-Liste angegeben:

d) *Distanz und Ranking nach einem neuen Input*: Die Klassifikation der verschiedenen Objekte wird zum Verständnis sehr vereinfacht ausgedrückt: Die Attribute des neuen Objekts werden als *neuer* Eingabevektor der Gewichtsmatrix übergeben (unter Anwendung der eingestellten Aktivierungsfunktion). Anschließend wird geprüft, welche Aktivierungswerte sich bei den *gelernten* Objekten durch diese neue Eingabe ergeben.

In der Rangliste wird das Objekt der semantischen Matrix an erster Stelle aufgeführt, das am stärksten durch einen Eingabevektor aktiviert wird. In der Distanzliste steht an erster Stelle das Objekt der semantischen Matrix, das die kleinste Differenz (Euklidische Distanz) zu dem Eingabevektor aufweist.

Entsprechend sehen die Ergebnisse für das kleine Beispiel aus (Abb. 2.7).

Da nur ein einziges Attribut in dem Eingabevektor einen numerischen Wert erhält, sind weder die Aktivierungen in der Rangliste sehr hoch, noch die Distanzen sehr klein,

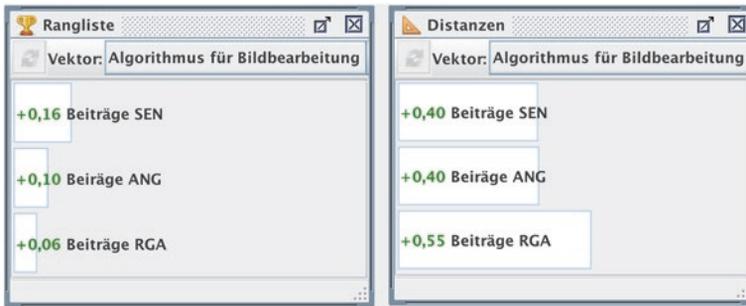


Abb. 2.7 Rangliste (links) und Distanzen (rechts)

da insgesamt eine starke Abweichung zwischen den gelernten Vektoren und dem neuen Objekt vorhanden ist.

Es sei hier bereits angemerkt, dass es durchaus eine Herausforderung ist zu entscheiden, welche Ergebnisse für die eigene Problemstellung wesentlich sind. In dem kleinen Beispiel ist an erster Stelle sowohl in der Rangliste als auch in den Distanzen dasselbe (und korrekte) Ergebnis. Es kann jedoch auch Probleme geben, bei denen nur die Rangliste zu berücksichtigen ist und andere, bei denen nur die Distanzen relevant sind. Ein erster Ansatz zur formalen Bestimmung wird in Kap. 26 vorgestellt.

In den Beiträgen wird jeweils problemorientiert entschieden, ob die Ranglisten oder die Distanzen zu berücksichtigen sind. Für manche Problemstellungen, insbesondere wenn es um eine Entscheidungsunterstützung in kritischen Bereichen geht, ist es unabdingbar, dass beide Ergebnisse übereinstimmend sind. Ist dies nicht der Fall, muss ein menschlicher Experte hinzugezogen werden.

Methodische Erweiterungen Auf zwei *methodische* Erweiterungen des SEN-Algorithmus muss noch verwiesen werden:

- a) Will man die Relevanz von einem oder mehreren Attributen besonders hervorheben, dann werden diese in der semantischen Matrix durch einen besonderen, gewöhnlich erhöhten Wert charakterisiert – der in der allgemeinen Einleitung und in Gl. 2.10 bereits erwähnte *cue validity factor (cvf)*.
- b) Die zweite Erweiterung besteht in der Bestimmung eines oder mehrerer Objekte als *Referenztypen*. Damit ist gemeint, dass die Objekte insgesamt klassifiziert werden in ihrer Relation zu bestimmten bereits vorhandenen oder zusätzlich konstruierten Objekten. Die Grundidee stammt ebenfalls von Rosch (1973), die den Lernprozess von Kindern dadurch charakterisierte, dass Kinder neue Wahrnehmungen an bereits vorhandenen Wahrnehmungen orientieren und z. B. eine neu wahrgenommene Katze an ihrer Ähnlichkeit zu bereits bekannten Katzen *als Katze* erkennen.³

³Rosch spricht hier von *Prototypen*. Da dieser Begriff in den technischen Wissenschaften bereits mit anderen Bedeutungen verwendet wird, haben wir hier den Begriff „Referenztypus“ eingeführt.

Werden Referenztypen als Eingabevektoren in SEN visualisiert, dann handelt es sich um den „Reference Type Centered Modus“ (Klüver et al. loc. cit.).

Dies kann abschließend zu dieser Einleitung an einem realen Beispiel verdeutlicht werden, das wir bereits in Klüver et al. 2012 dokumentiert haben: Ein ehemaliger Student von uns hatte vor etlichen Jahren in seiner Firma den Auftrag, einen möglichst geeigneten Standort für eine neue Offshore Windkraftanlage zu bestimmen. Dazu hatte er eine Liste von möglichen Standorten erhalten mit Attributen wie Entfernung vom Festland, durchschnittliche Wassertiefe, Meeresströmungen, nächste Anschlüsse an das allgemeine Stromnetz etc. Aus diesen Daten und durch Expertenbefragungen konstruierte er die semantische Matrix für sein SEN Modell.

Im nächsten methodischen Schritt konstruierte er einen fiktiven Standort, der alle gewünschten Eigenschaften optimal erfüllte, aber den es nicht real gibt. Man kann hier im Sinne des großen theoretischen Soziologen Max Weber von einem *Idealtypus* sprechen. Dies neue Objekt wurde nun als Referenztypus in das SEN Modell eingesetzt und zwar gemäß der Methode des Zentrummodus (c) im Zentrum des Modells. Die bereits eingesetzten realen Standorte wurden dann vom Algorithmus zum Zentrum „gezogen“ – je näher ein Objekt dem Zentrum ist, desto besser geeignet ist der entsprechende Standort.



Im Jahr 2019 wurde dieser Standort in Betrieb genommen⁴ – eine für uns sehr erfreuliche Information. Mit SEN lässt sich also auch eine Ordnung nach Qualitätskriterien herstellen und visuell verdeutlichen.

Damit sind wir bereits in der ersten praktischen Anwendung von SEN angelangt. Der eigentliche Hauptteil, nämlich die verschiedenen praktischen Beispiele, soll damit auch beginnen.

⁴https://de.wikipedia.org/wiki/Liste_der_Offshore-Windparks

2.3 Anwendungsgebiete des SEN

Die Einsatzgebiete des SEN sind zwischenzeitlich sehr vielfältig. Wurde SEN ursprünglich primär zur Modellbildung und Klassifikation verwendet, ist es zwischenzeitlich möglich, komplexe Daten zu clustern und zu analysieren, SEN als Entscheidungsunterstützung einzusetzen oder zur Bildbearbeitung. Selbst zur Analyse und Klassifikation akustischer Signale gibt es bereits einen Prototypen.

Ein Vorteil von SEN besteht darin, dass dieses neuronale Netzwerk gemäß der Aufteilung in Kap. 1 sowohl als Methode der Künstlichen Intelligenz als auch des Maschinellen Lernens eingesetzt werden kann. Dies wird in den folgenden Kapiteln anschaulich demonstriert.

Zunächst werden Modelle vorgestellt, in denen typische Probleme aufgegriffen werden, die im Kontext der Softwareentwicklung auftreten: Die Auswahl eines Vorgehensmodells, die Sicherung bzw. Verbesserung der Qualität, Aufwandschätzungen oder die Identifikation von Aufwandstreiber. Darüber hinaus geht es um eine Entscheidungsunterstützung für die Auswahl von Softwarekomponenten, oder für die Wahl eines individuell geeigneten Führungsstils. Es wird gezeigt, dass auch Begriffe wie „Effektivität“ bei Beratern oder „Effizienz“ in agilen Teams durch die entwickelten Modelle präzisiert werden können.

Anschließend werden zwei Wahlen aus unterschiedlicher Perspektiven analysiert: Zunächst geht es um die Rekonstruktion der USA-Wahlen im Jahr 2016 und darauf folgend um die Prognose der Bundestagswahlen im Jahre 2017. Diese Modelle zeigen, dass die ausschließliche Berücksichtigung der Wahlprognosen nicht hilfreich ist, um nachzuvollziehen, warum die Wahlentscheidungen so ausgefallen sind. Werden die Modelle erweitert und zusätzliche Faktoren berücksichtigt, in dem einen Fall die Umfragen in den USA und im anderen Fall die Aktivitäten in den Social Media, so schlagen die SEN-Ergebnisse sogar die Wahlprognosen.

Die nächsten vier Beiträge enthalten sehr heterogene Problemstellungen: Die Entscheidung für eine Start- und Landebahn am Frankfurter Flughafen in Abhängigkeit von Wetterprognosen, die Identifikation von Aquisizeilen in der Stahlbranche und von Angriffstools in Voice Over IP-Systemen und schließlich der Abweichungen in handschriftlich eingegebenen Daten in Abrechnungen.

Die letzten beiden Beiträge in diesem Teil zeigen die Möglichkeiten auf, SEN für die Bildbearbeitung einzusetzen.

Literatur

- Klüver C, Klüver J, Schmidt J (2012) Modellierung komplexer Prozesse durch naturanaloge Verfahren. *Soft Computing und verwandte Techniken*, 2. Aufl. Springer Vieweg, Wiesbaden
- Klüver C (2016) Steering Clustering of Medical Data in a Self-Enforcing Network (SEN) with a Cue Validity Factor. *IEEE Symposium Series on Computational Intelligence*, Athen 1–8. <https://doi.org/10.1109/SSCI.2016.7849883>
- Rosch E (1973) Natural Categories. *Cogn Psychol* 4:328–350



Bewertung und Auswahl von Vorgehensmodellen im IT-Projektmanagement – Ein Ansatz für die Unternehmenspraxis

3

Christoph Albers

Zusammenfassung

Die Auswahl eines geeigneten Vorgehensmodells für ein Projekt ist für viele IT-Manager ein wiederkehrendes Problem. Zur Unterstützung existieren vielfältige theoretisch-orientierte Ansätze, die jedoch unterschiedliche Defizite mit sich bringen, so dass eine erfolgreiche Überführung in die Unternehmenspraxis bisweilen nicht erfolgt ist. Dieser Beitrag stellt einen neuen Ansatz zur Vorgehensmodellauswahl vor, welcher auf einem künstlichen neuronalen Netzwerk, dem Self-Enforcing Network, basiert und zugleich die Ergebnisse einer Expertenbefragung einbezieht. Anhand zweier Beispiele wird vorgestellt, wie dieser Ansatz bei der Auswahl von agilen und sogar hybriden Vorgehensmodellen Entscheidern durch ein formales Modell einen Mehrwert erbringen kann.

Schlüsselwörter

Self-Enforcing Network · Projektmanagement · Vorgehensmodell · IT-Management · IT-Governance

C. Albers (✉)

Commercial Aviation Platforms – Group IT, TUI Infotec GmbH, Hannover, Deutschland

E-Mail: christoph.albers@tui.com

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2021

C. Klüver und J. Klüver (Hrsg.), *Neue Algorithmen für praktische Probleme*,

https://doi.org/10.1007/978-3-658-32587-9_3

21

3.1 Status Quo: Die Auswahl von Vorgehensmodellen in der Unternehmenspraxis

Entscheidungsträger in der Informationstechnologie (IT) wie Chief Information Officer, Programm-, Projekt- und Produktmanager stehen regelmäßig vor der Entscheidung für ein Projekt, ein Programm, eine Produktentwicklung oder auch als organisatorische Vorgabe ein geeignetes Vorgehensmodell (VM) auszuwählen. Dabei stoßen Denkansätze wie die des klassischen Projektmanagements gegen die des agilen Produktmanagements oder Verfechter hybrider Verfahren. Doch nicht nur die hohe Anzahl der verfügbaren Vorgehensmodelle an sich macht die Entscheidungsfindung hochgradig komplex. Vorgehensmodelle können zudem miteinander kombiniert werden, so z. B. PRINCE2 und Scrum, oder durch sogenanntes Tailoring an den Kontext des Projektes angepasst werden.

Das Ziel dieses Beitrags besteht darin, einen Ansatz vorzustellen, welcher dem Wunsch nach einer Unterstützung bei der Auswahl eines Vorgehensmodells im Kontext des IT-Projektmanagements nachkommt.

Hierzu wird ein Entscheidungsmodell mit einem entsprechenden Entscheidungsfeld bestehend aus den Handlungsalternativen (Vorgehensmodelle), den Bewertungskriterien und den Bewertungen erarbeitet. Die Bewertung erfolgte durch eine Expertenbefragung und die Ergebnisse wurden in die semantische Matrix des Self-Enforcing Network (SEN) überführt (Abschn. 3.3.1).

Dieser grundlegenden Arbeit folgend wird anhand zweier Beispiele aufgezeigt, wie eine Auswahl eines VM anhand eines exemplarischen Projektes durch das als SEN implementierte Entscheidungsmodell unterstützt werden kann.

Abschließend wird das Ergebnis dieser Beispiele kritisch betrachtet und diskutiert sowie ein Ausblick auf weiterführende Arbeiten an diesem Ansatz geboten.

3.1.1 Einordnung und Bedeutung von Vorgehensmodellen im IT-Projektmanagement

Dass die Fragestellung nach dem geeigneten Vorgehensmodell für ein Projekt nicht leichtfertig getroffen werden sollte, kann aus der nachfolgenden Definition und der Bedeutung von VMs für das IT-Projektmanagement abgeleitet werden:

Vorgehensmodelle im Kontext des IT-Projektmanagements geben einen übergeordneten Ablauf für Projekte vor. Sie stellen Prozesse und Methoden zur Verfügung, wiederkehrende Tätigkeiten des Projektmanagements in gleicher Art und Weise umsetzen zu können (Broy und Kuhrmann 2013). In dieser Funktion können die VM als systematisches Werkzeug der Projektplanung-, Durchführung und –Überwachung aufgefasst werden (Wieczorrek und Mertens 2011), welche ein einzelnes Projekt in einen standardisierten Rahmen einbetten. Ferner geben Vorgehensmodelle die Schnittstellen