daniel BASLER

Neuronale Netze mit C# programmieren

Mit praktischen Beispielen für Machine Learning im Unternehmenseinsatz

Beispielcode unter
plus.hanser-fachbuch.de

HANSER

HANSER

Daniel Basler

Neuronale Netze mit C# programmieren

Mit praktischen Beispielen für Machine Learning im Unternehmenseinsatz

Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial.

Geben Sie auf **plus.hanser-fachbuch.de** einfach diesen Code ein:

plus-5db90-her61

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

© 2021 Carl Hanser Verlag München, *www.hanser-fachbuch.de*
Copy editing: Walter Saumweber, Ratingen
Umschlagdesign: Marc Müller-Bremer, *www.rebranding.de*, München
Umschlagrealisation: Max Kostopoulos
Titelmotiv: © *istockphoto.com*/Artystarty
Layout: Manuela Treindl, Fürth
Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Print-ISBN: 978-3-446-46229-8
E-Book-ISBN: 978-3-446-46426-1

E-Pub-ISBN: 978-3-446-46635-7

Inhalt

2.5 Lernformen

2.5.1 Überwachtes Lernen

2.5.2 Unüberwachtes Lernen

2.5.3 Semi-überwachtes Lernen

2.5.4 Verstärkendes Lernen

2.6 Machine-Learning-Algorithmen

2.6.1 *k*-Nearest-Neighbour

2.6.2 Support Vector Machine

2.6.3 Entscheidungsbäume

2.6.4 Decision Tree und Random-Forest

2.6.5 Clustering

2.6.5.1 K-Means Clustering

2.6.5.2 EM-Clustering

2.6.5.3 Hierarchische Clusteranalyse

2.7 Training und Validierung des ML-Modells

2.8 Das einfache neuronale Netz

Vorwort

Neuronale Netze sind seit einiger Zeit überall im Gespräch. Als Softwareentwickler stellt man fest, dass es zurzeit keinen anderen Bereich in der Softwareentwicklung gibt, der sich so rasant verändert und weiterentwickelt.

Eine attraktive Alternative zu Python für den Entwurf von neuronalen Netzen ist eine modulare und objektorientierte Programmiersprache wie Microsoft C#. Die Objektorientierung bietet den Vorteil, dass sich sehr gut modulare Machine-Learning-Modelle entwerfen lassen, die konfigurierbar, erweiterbar und wiederverwendbar sind.

Dieses Buch richtet sich an C#-Entwickler, die einen möglichst umfassenden Blick über neuronale Netze erlangen wollen. Es möchte Sie beim Kennenlernen, Experimentieren und Arbeiten mit neuronalen Netzen und Machine-Learning-Modellen anleiten und unterstützen. Dabei wendet es sich an im Umgang mit neuronalen Netzen unerfahrene Programmierer.

Sie sollten über Grundkenntnisse in der Programmierung mit C# verfügen, sodass Begriffe wie Variablen, Schleifen etc. Ihnen vertraut sind. Wegen des mathematischen Anteils müssen Sie sich keine Sorgen machen. Um die Buchinhalte zu verstehen,

sind wirklich nur gute Kenntnisse in dem Konzept der linearen Algebra erforderlich.

Insgesamt erhebt das Buch auch keinen Anspruch auf Vollständigkeit. Es verzichtet auf tiefgehende mathematische und programmiertechnische Details, die nicht wirklich notwendig sind, um die Programmierung von neuronalen Netzen und Machine Learning zu verstehen.

Anhand von Anwendungsbeispielen lernen Sie neuronale Netze und Machine-Learning-Modelle zu entwickeln. Sie lernen auf diese Weise dynamische Datenstrukturen, Feedforward-Netze, Backpropagation-Algorithmen sowie Convolutional Neural Networks und Natural Language Processing kennen. Das Buch möchte Ihnen die Leistungsvielfalt neuronaler Netze vermitteln und Ihnen helfen diese in eigenen Programmierprojekten zu nutzen.

Den Mitarbeiterinnen und Mitarbeitern des Hanser-Verlages, besonders Frau Sylvia Hasselbach, danke ich für die Sorgfalt und Unterstützung bei der Veröffentlichung dieses Buches.

Ihnen, liebe Leserin und lieber Leser, wünsche ich viel Freude und Erfolg beim Kennenlernen und Arbeiten mit neuronalen Netzen mit C#.

Daniel Basler

Herford, April 2021

Aufbau des Buches

Die wichtigsten Begriffe und Konzepte werden gleich in den ersten beiden Kapiteln erläutert. Die weiteren Kapitel sind thematisch so aufgebaut, dass man das Buch von vorne nach hinten durcharbeiten kann.

Nachfolgend erhalten Sie einen Überblick über den Inhalt des Buches:

- **Kapitel 1** führt in das Thema künstliche Intelligenz ein. Dabei geht es um die Anwendungsgebiete, den Aufbau von neuronalen Netzen und das Neuron als Grundbaustein.
- **Kapitel 2** beschreibt die Konzepte und Methoden von Machine Learning. Es nennt wichtige Algorithmen und zeigt ein erstes einfaches neuronales Netz als Programmierbeispiel.
- **Kapitel 3** beschreibt Schritt für Schritt den Aufbau eines künstlichen neuronalen Netzes in C# und geht auf die wichtigsten Mathematik-Grundlagen ein. Anhand von Aufgabenstellungen werden mehrschichtige neuronale Netze entwickelt. Des Weiteren beschäftigen Sie sich in diesem Kapitel mit Lernalgorithmen und der Programmierung des Backpropagation-Algorithmus.

- **Kapitel 4** zeigt das Training von neuronalen Netzen. Sie lernen hier unterschiedliche Lernprozesse kennen und implementieren Trainingsmethoden in ein neuronales Netz.
- **Kapitel 5** führt Sie weiter zu den sogenannten Recurrent Neural Networks (RNN), die hauptsächlich in der Verarbeitung von Textsequenzen oder Zeitreihen eingesetzt werden. Sie lernen, wie die einzelnen Schichten in einem RNN aufgebaut sind.
- **Kapitel 6** erläutert ausführlich den Aufbau von Convolutional Neural Networks (CNN), die vornehmlich für die Verarbeitung von Bild- und Audiodateien eingesetzt werden. Ein CNN ist ein spezielles mehrschichtiges Feedforward-Netz.
- **Kapitel 7** beschäftigt sich mit den im Moment aktuellen Machine Learning Frameworks, die Sie bei der Entwicklung von Machine-Learning-Modellen in C# unterstützen können.
- **Kapitel 8** beschreibt ein weiteres nützliches ML-Framework im .NET Umfeld. Der sogenannte SciSharp-Technologie-Stack ermöglicht es, ML-Modelle für TensorFlow mit C# zu erstellen. Des Weiteren werden in diesem Kapitel die Frameworks Keras.NET und NeuralNetwork.NET vorgestellt.
- **Kapitel 9** gibt einen Einblick in Machine Learning as a Service und stellt Amazon Lex und Azure Cognitive Services an einem Anwendungsbeispiel näher vor.
- **Kapitel 10** setzt die in den vorherigen Kapiteln aufgezeigten Methoden in einzelnen Beispielanwendungen für Zeitreihenanalyse, Bildklassifikation, Objekterkennung und Natural Language Processing um.

Programmbeispiele

Um den Praxisbezug zu gewährleisten, wird gezeigt, wie die beschriebenen Themen in C#-Programme umgesetzt werden. In den meisten Fällen wird der vollständige Beispielcode abgebildet, um die Programmierung in C# zu verdeutlichen. Sie können die Buchbeispiele komplett von meinem GitHub-Repository herunterladen (*https://github.com/DanielBasler/NeuralNetwork*) oder von der Plus.Hanser-Webseite:

Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel die Code-Beispiele aus dem Buch. Geben Sie auf *plus.hanser-fachbuch.de* einfach diesen Code ein:

```
plus-5db90-her61
```

Zielsetzung des Buches

Mein Ziel ist es, Ihnen ein solides Fundament für das Entwerfen und Entwickeln von neuronalen Netzen und Machine-Learning-Modellen an die Hand zu geben, unterstützt von praktischen Beispielen. Des Weiteren möchte ich, dass Sie ein Gefühl für den Programmier-Aufwand von neuronalen Netzen und deren Leistung bekommen und die Vielfalt der Einsatzmöglichkeiten aber auch der Anforderungen kennenlernen.

Alle hier verwendeten Softwaretools, mit Ausnahme der KI-Services, sind kostenlos und Open Source. Die Beispiele lassen

sich mit Visual Studio Community Version 2019 entwickeln und auf einem „normalen PC“ ohne besondere Hardware-Power erstellen und ausführen.

1 Künstliche Intelligenz

„Okay Google“, „Alexa, spiel bitte ‚Eight days a week‘ von den Beatles“ oder „hey Siri“… kennen Sie vermutlich alle. Die sogenannte künstliche Intelligenz, kurz nur noch als KI bezeichnet, ist schon längst in unserem Alltag angekommen und dringt in immer mehr Bereiche vor.

Das heißt, wir sind alle mittendrin, ob es um Sprachassistenten als Helfer auf dem Smartphone, im Auto oder zu Hause geht, die digitale Vernetzung in allen Lebensbereichen schreitet mit unglaublicher Geschwindigkeit voran. Mit der digitalen Transformation, durch die auch die KI allgegenwärtig wird bzw. ist, müssen sich auch Unternehmen auf eine Umstrukturierung und gegebenenfalls sogar auf eine Neuorientierung ihrer Geschäftsprozesse einstellen. Heute müssen Sie sich als Entwickler mit Begriffen wie digitaler Zwilling, Product Performance Management und digitale Fabrik auseinandersetzen. So stellt zum Beispiel das Zusammenspiel von Robotic Process Automation (RPA) und künstlicher Intelligenz ganz neue Symbiosen im Bereich der Automation dar.

In der Diskussion über KI tauchen heute sehr viele verschiedene Begriffen auf. Es ist die Rede von Machine Learning, neuronalen Netzen, Representation Learning, Natural Language Processing

(NLP) oder auch Deep Learning. Selbst Begriffe wie Big Data und Data Science werden in die Runde geworfen. Sie sehen also, wer sich mit KI befasst, wird sehr schnell mit einem Begriffswirrwarr von Schlagwörtern überzogen, die durchaus für Verwirrung sorgen können.

Man kann KI als Überbegriff sehen, unter dem unterschiedliche Techniken und Bezeichnungen versammelt sind, und bevor wir uns dem Hauptthema des Buches – der Programmierung neuronale Netze – widmen, möchte ich einige Begriffe und Technologien klären, die in diesem Buch benutzt werden.

1.1 Grundlagen

Dabei ist KI keine neue Wissenschaft oder Technologie. Die Anfänge der KI-Forschung gehen bis in die 1950er-Jahre zurück, in denen Alan Turing den Aufsatz „Computing Machinery and Intelligence“ [1] vorgelegt hat. Auf Turing geht auch der nach ihm benannte Turing-Test zurück, der dazu dient, zu unterscheiden, ob eine Maschine ein gleichwertiges Denkvermögen aufweist wie ein Mensch oder nicht.

Zum ersten Mal wurde der Begriff „artifical intelligence“ von John McCarthy [2] verwendet bzw. geprägt. Laut McCarthy ist KI eine Informations- und Ingenieurswissenschaft, die dem Herstellen „intelligenter“ Maschinen und speziellen intelligenten Computerprogrammen gewidmet ist. Für McCarthy besteht der rechnerische Teil der Intelligenz in der Fähigkeit, die Ziele in der Welt zu erreichen. Das heißt für ihn, ein Computer soll so gebaut oder programmiert werden, dass er eigenständig Programme