

---

# Neuronale Netze selbst programmieren

*Ein verständlicher Einstieg mit Python*

*Tariq Rashid*

*Deutsche Übersetzung von  
Frank Langenau*

**O'REILLY®**

Tariq Rashid

Lektorat: Alexandra Follenius

Übersetzung: Frank Langenau

Korrektur: Sibylle Feldmann

Herstellung: Susanne Bröckelmann

Umschlaggestaltung: Michael Oréal, [www.oreal.de](http://www.oreal.de)

Satz: III-satz, [www.drei-satz.de](http://www.drei-satz.de)

Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

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.

ISBN:

Print 978-3-96009-043-4

PDF 978-3-96010-102-4

ePub 978-3-96010-103-1

mobi 978-3-96010-104-8

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

1. Auflage 2017

Copyright © 2016 by Tariq Rashid

Title of the English original: Make Your Own Neural Network

ISBN 978-1530826605

Translation Copyright © 2017 by dpunkt.verlag GmbH. All rights reserved.

Wiebinger Weg 17

69123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Die Informationen in diesem Buch wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Verlag, Autoren und Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene Fehler und deren Folgen.

5 4 3 2 1 0

<b>Einführung</b>	<b>IX</b>
<b>1 Wie neuronale Netze arbeiten</b>	<b>1</b>
Leicht für mich – schwer für dich	1
Eine einfache Vorhersagemaschine	3
Klassifizieren unterscheidet sich nicht sehr vom Vorhersagen	8
Einen einfachen Klassifizierer trainieren	14
Manchmal ist ein Klassifizierer nicht genug	24
Neuronen – die Rechenmaschinen der Natur	30
Signalen durch ein neuronales Netz folgen	39
Matrizenmultiplikation ist nützlich – ehrlich!	44
Beispiel: Ein dreischichtiges Netz mit Matrizenmultiplikation	51
Gewichte von mehr als einem Knoten lernen	59
Fehler von mehreren Ausgabeknoten zurückführen	61
Fehler auf mehrere Schichten zurückführen	63
Backpropagierung von Fehlern mit Matrizenmultiplikation	67
Wie aktualisieren wir eigentlich die Gewichte?	70
Gewichtsaktualisierung am konkreten Beispiel	88
Die Daten vorbereiten	89
Eingaben	90
Ausgaben	91
Zufällige Anfangswerte	92
<b>2 Do it yourself mit Python</b>	<b>95</b>
Python	95
Interaktives Python = IPython	96
Ein sehr sanfter Start mit Python	97
Notebooks	97
Einfaches Python	99

Arbeiten automatisieren .....	101
Kommentare .....	104
Funktionen .....	104
Arrays .....	107
Arrays grafisch darstellen .....	110
Objekte .....	112
Neuronales Netz mit Python .....	118
Der Gerüstcode .....	118
Das Netz initialisieren .....	119
Gewichte – das Herz des Netzes .....	121
Optional: differenzierte Initialisierung der Gewichte .....	123
Das Netz abfragen .....	124
Der aktuelle Stand des Codes .....	126
Das Netz trainieren .....	129
Der vollständige Code für das neuronale Netz .....	132
Die MNIST-Datenbank mit handgeschriebenen Ziffern .....	133
Die MNIST-Trainingsdaten vorbereiten .....	141
Das Netz testen .....	148
Mit sämtlichen Datensätzen trainieren und testen .....	152
Verbesserungen: Optimieren der Lernrate .....	153
Verbesserungen: Mehrere Läufe .....	155
Die Gestalt des Netzes ändern .....	157
Gute Arbeit! .....	159
Der endgültige Code .....	160
<b>3 Just for fun: Das neuronale Netz tunen .....</b>	<b>165</b>
Ihre eigene Handschrift .....	165
Das Gedächtnis eines neuronalen Netzes .....	168
Geheimnisvolle Blackbox .....	168
Rückwärtsabfrage .....	169
Die Kennung »0« .....	170
Weitere Hirnscans .....	171
Neue Trainingsdaten erzeugen: Drehungen .....	173
<b>Epilog .....</b>	<b>177</b>
<b>A Eine leicht verständliche Einführung in die Analysis .....</b>	<b>179</b>
Eine Gerade .....	180
Eine schräg verlaufende Gerade .....	182
Eine gekrümmte Kurve .....	184
Analysis per Hand .....	186
Analysis nicht per Hand .....	188

Analysis, ohne Graphen zu zeichnen .....	191
Muster .....	194
Funktionen von Funktionen .....	196
Sie können Analysis betreiben! .....	199
<b>B Das Ganze mit einem Raspberry Pi .....</b>	<b>201</b>
IPython installieren .....	202
Vergewissern, dass alles funktioniert .....	209
Ein neuronales Netz trainieren und testen .....	210
Erfolg für Raspberry Pi! .....	211
<b>Index .....</b>	<b>213</b>