

Auf einen Blick

1	Einleitung	19
2	Das minimale Starterkit für die Entwicklung von neuronalen Netzen mit Python	47
3	Ein einfaches neuronales Netz	69
4	Lernen im einfachen Netz	101
5	Mehrschichtige neuronale Netze	139
6	Lernen im mehrschichtigen Netz	161
7	Convolutional Neural Networks	197
8	Programmierung von Convolutional Neural Networks mit TensorFlow 2	221
9	Vom Hirn zum Netz	251
10	Die Evolution der neuronalen Netze	265
11	Der Machine-Learning-Prozess	289
12	Lernverfahren	325
13	Anwendungsbereiche und Praxisbeispiele	367

Inhalt

Vorwort zur 2. Auflage	13
Materialien zum Buch	14
Vorwort	15

1 Einleitung 19

1.1 Wozu neuronale Netze?	19
1.2 Über dieses Buch	20
1.3 Der Inhalt kompakt	22
1.4 Ist diese Biene eine Königin?	25
1.5 Ein künstliches neuronales Netz für den Bienenstaat	26
1.6 Von der Biologie zum künstlichen Neuron	31
1.6.1 Das biologische Neuron und seine technische Kopie	32
1.6.2 Das künstliche Neuron und seine Elemente	33
1.7 Einordnung und der Rest	36
1.7.1 Big Picture	36
1.7.2 Artificial Intelligence (künstliche Intelligenz)	37
1.7.3 Geschichte	38
1.7.4 Machine Learning (maschinelles Lernen)	40
1.7.5 Deep Neural Networks	41
1.8 Zusammenfassung	43
1.9 Referenzen	44

Teil I Up and running

2 Das minimale Starterkit für die Entwicklung von neuronalen Netzen mit Python 47

2.1 Die technische Entwicklungsumgebung	47
2.1.1 Die Anaconda-Distribution	47
2.1.2 Unser Cockpit: Jupyter Notebook	52

2.1.3	Wichtige Python-Module	64
2.1.4	Jupyter-Notebook-Cloud-Ressourcen	66
2.2	Zusammenfassung	67
3	Ein einfaches neuronales Netz	69
3.1	Vorgeschichte	69
3.2	Her mit dem neuronalen Netz!	69
3.3	Neuron-Zoom-in	73
3.4	Stufenfunktion	78
3.5	Perceptron	80
3.6	Punkte im Raum – Vektorrepräsentation	81
3.6.1	Aufgabe: Werte vervollständigen	82
3.6.2	Aufgabe: Iris-Datensatz als Scatter-Plot ausgeben	85
3.7	Horizontal und vertikal – Spalten- und Zeilenschreibweise	88
3.7.1	Aufgabe: Ermittlung des Skalarprodukts mithilfe von NumPy	89
3.8	Die gewichtete Summe	91
3.9	Schritt für Schritt – Stufenfunktionen	91
3.10	Die gewichtete Summe reloaded	92
3.11	Alles zusammen	93
3.12	Aufgabe: Roboterschutz	96
3.13	Zusammenfassung	99
3.14	Referenzen	99
4	Lernen im einfachen Netz	101
4.1	Vorgeschichte: Man lässt planen	101
4.2	Lernen im Python-Code	102
4.3	Perceptron-Lernen	103
4.4	Trenngerade für einen Lernschritt	106
4.5	Perceptron-Lernalgorithmus	108
4.6	Die Trenngeraden bzw. Hyperplanes oder auch Hyperebenen für das Beispiel	113

4.7	scikit-learn-kompatibler Estimator	116
4.8	scikit-learn-Perceptron-Estimator	123
4.9	Adaline	126
4.10	Zusammenfassung	136
4.11	Referenzen	137
5	Mehrschichtige neuronale Netze	139
5.1	Ein echtes Problem	139
5.2	XOR kann man lösen	141
5.3	Vorbereitungen für den Start	147
5.4	Der Plan für die Umsetzung	149
5.5	Das Setup (»class«)	150
5.6	Die Initialisierung (»__init__«)	152
5.7	Was für zwischendurch (»print«)	154
5.8	Die Auswertung (»predict«)	155
5.9	Die Verwendung	157
5.10	Zusammenfassung	159
6	Lernen im mehrschichtigen Netz	161
6.1	Wie misst man einen Fehler?	161
6.2	Gradientenabstieg an einem Beispiel	163
6.2.1	Gradientenabstieg – die Idee	163
6.2.2	Algorithmus für den Gradientenabstieg	165
6.3	Ein Netz aus sigmoiden Neuronen	172
6.4	Der coole Algorithmus mit Vorwärts-Delta und Rückwärts-Propagation	174
6.4.1	»__init__«-Methode	174
6.4.2	»predict«-Methode	177
6.4.3	»fit«-Methode	182
6.4.4	»plot«-Methode	184
6.4.5	Alles im Konzert	185
6.5	Ein »fit«-Durchlauf	187
6.5.1	Initialisierung	189

- 6.5.2 Forward 190
- 6.5.3 Output 191
- 6.5.4 Hidden 192
- 6.5.5 Delta W_{kj} 194
- 6.5.6 Delta W_{ji} 195
- 6.5.7 W_{ji} 195
- 6.5.8 W_{kj} 195
- 6.6 Zusammenfassung** 196
- 6.7 Referenz** 196

7 Convolutional Neural Networks 197

- 7.1 Aufbau eines CNN** 199
- 7.2 Der Kodierungsblock** 200
 - 7.2.1 Convolutional Layer 200
 - 7.2.2 Activation Function 203
 - 7.2.3 Pooling Layer 204
 - 7.2.4 Überlappen, ausfüllen und Schrittlänge 205
- 7.3 Der Prädiktionsblock** 207
 - 7.3.1 Flatten 207
 - 7.3.2 Softmax 208
- 7.4 Trainieren von Convolutional Neural Networks** 209
 - 7.4.1 Das Problem der explodierenden/verschwindenden Gradienten 210
 - 7.4.2 Das Optimierungsverfahren 214
 - 7.4.3 Verhindern von Overfitting 216
- 7.5 Zusammenfassung** 218
- 7.6 Referenzen** 219

8 Programmierung von Convolutional Neural Networks mit TensorFlow 2 221

- 8.1 Convolutional Networks zur Handschriftenerkennung** 221
 - 8.1.1 Der Datensatz 221
 - 8.1.2 Ein einfaches CNN 225
 - 8.1.3 Die Ergebnisse 230

- 8.2 Transfer Learning mit Convolutional Neural Networks** 237
 - 8.2.1 Das vortrainierte Netzwerk 239
 - 8.2.2 Datenvorbereitung 240
 - 8.2.3 Das vortrainierte Netz 241
 - 8.2.4 Die Ergebnisse 244
- 8.3 Zusammenfassung** 246
- 8.4 Referenzen** 247

Teil II Deep Dive

9 Vom Hirn zum Netz 251

- 9.1 Ihr Gehirn in Aktion** 251
- 9.2 Das Nervensystem** 252
- 9.3 Das Gehirn** 253
 - 9.3.1 Die Teile 253
 - 9.3.2 Ein Ausschnitt 254
- 9.4 Neuronen und Gliazellen** 255
- 9.5 Eine Übertragung im Detail** 257
- 9.6 Darstellung von Zellen und Netzen** 260
- 9.7 Zusammenfassung** 262
- 9.8 Referenzen** 263

10 Die Evolution der neuronalen Netze 265

- 10.1 1940er** 265
 - 10.1.1 1943: McCulloch-Pitts Neurons 266
 - 10.1.2 1949: Donald Hebb 267
- 10.2 1950er** 268
 - 10.2.1 1951: Marvin Minsky und Dean Edmonds – SNARC 268
 - 10.2.2 1956: Artificial Intelligence 268
 - 10.2.3 1957: Rosenblatts Perceptron 268
 - 10.2.4 1959: Bernard Widrow und Marcian Hoff – Adaline und Madaline 269
- 10.3 1960er** 270
 - 10.3.1 1969: Marvin Minsky und Seymour Papert 270

- 10.4 1970er** 270
 - 10.4.1 1972: Kohonen – assoziativer Memory 270
 - 10.4.2 1973: Lighthill Report 270
 - 10.4.3 1974: Backpropagation 271
- 10.5 1980er** 271
 - 10.5.1 1980: Fukushimas Neocognitron 271
 - 10.5.2 1982: John Hopfield 273
 - 10.5.3 1982: Kohonens SOM 283
 - 10.5.4 1986: Backpropagation 284
 - 10.5.5 1987: NN-Konferenz 284
- 10.6 1990er** 284
 - 10.6.1 1997 Sepp Hochreiter und Jürgen Schmidhuber – LSTM 284
- 10.7 2000er** 285
 - 10.7.1 2006: Geoffrey Hinton et al. 285
- 10.8 2010er** 285
 - 10.8.1 2014: Ian J. Goodfellow et al. – Generative Adversarial Networks (GAN) 285
- 10.9 Zusammenfassung** 287
- 10.10 Referenzen** 288

11 Der Machine-Learning-Prozess 289

- 11.1 Das CRISP-DM-Modell** 289
 - 11.1.1 Geschäfts(prozess)-Verständnis 290
 - 11.1.2 Datenverständnis 291
 - 11.1.3 Datenvorbereitung 291
 - 11.1.4 Modellierung 292
 - 11.1.5 Evaluation 292
 - 11.1.6 Einsatz 293
- 11.2 Feature Engineering** 293
 - 11.2.1 Feature-Kodierung 295
 - 11.2.2 Feature-Extraktion 306
 - 11.2.3 Der Fluch der Dimensionalität 317
 - 11.2.4 Feature-Transformation 317
 - 11.2.5 Feature-Auswahl 322

- 11.3 Zusammenfassung** 324
- 11.4 Referenzen** 324

12 Lernverfahren 325

- 12.1 Lernstrategien** 325
 - 12.1.1 Überwachtes Lernen (Supervised Learning) 326
 - 12.1.2 Unüberwachtes Lernen (Unsupervised Learning) 330
 - 12.1.3 Verstärkendes Lernen (Reinforcement Learning) 344
 - 12.1.4 Teilüberwachtes Lernen (Semi-supervised Learning) 360
- 12.2 Werkzeuge** 361
 - 12.2.1 Confusion Matrix 361
 - 12.2.2 ROC-Curves 363
- 12.3 Zusammenfassung** 366
- 12.4 Referenzen** 366

13 Anwendungsbereiche und Praxisbeispiele 367

- 13.1 Warmup** 367
- 13.2 Bildklassifikation** 370
 - 13.2.1 Begriffsbestimmung 370
 - 13.2.2 Von Bienen und Hummeln 372
 - 13.2.3 (Vor-)Trainierte Netze 383
- 13.3 Erträumte Bilder** 391
 - 13.3.1 Der Algorithmus 392
 - 13.3.2 Die Implementierung 394
- 13.4 Zusammenfassung** 402
- 13.5 Referenzen** 402

- A Python kompakt** 403
- B Mathematik kompakt** 433
- C TensorFlow 2 und Keras** 455

- Index** 467