

Neuronale Netze programmieren mit Python

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhalt

Vorwort	13
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	32
1.6.1 Das biologische Neuron und seine technische Kopie	32
1.6.2 Das künstliche Neuron und seine Elemente	34
1.7 Einordnung und der Rest	36
1.7.1 Big Picture	37
1.7.2 Artificial Intelligence (künstliche Intelligenz)	37
1.7.3 Geschichte	39
1.7.4 Machine Learning (maschinelles Lernen)	41
1.7.5 Deep Neural Networks	42
1.7.6 Transformer Neural Networks	43
1.8 Zusammenfassung	45
1.9 Referenzen	45

Teil I Up and running

2 Das minimale Starterkit für die Entwicklung von neuronalen Netzen mit Python	49
2.1 Die technische Entwicklungsumgebung	49
2.1.1 Die Anaconda-Distribution	49
2.1.2 Unser Cockpit: Jupyter Notebook	54
2.1.3 Wichtige Python-Module	66

2.1.4	Die Google Colab-Plattform für Jupyter Notebooks	68
2.1.5	Weitere Jupyter Notebook-Cloud-Ressourcen	71
2.2	Zusammenfassung	72

3 Ein einfaches neuronales Netz 73

3.1	Vorgeschichte	73
3.2	Her mit dem neuronalen Netz!	73
3.3	Neuron-Zoom-in	77
3.4	Stufenfunktion	82
3.5	Perceptron	84
3.6	Punkte im Raum – Vektorrepräsentation	85
3.6.1	Aufgabe: Werte vervollständigen	86
3.6.2	Aufgabe: Den Iris-Datensatz als Scatter-Plot ausgeben	89
3.7	Horizontal und vertikal – Spalten- und Zeilenschreibweise	92
3.7.1	Aufgabe: Das Skalarprodukt mithilfe von NumPy ermitteln	93
3.8	Die gewichtete Summe	95
3.9	Schritt für Schritt – Stufenfunktionen	95
3.10	Die gewichtete Summe reloaded	96
3.11	Alles zusammen	97
3.12	Aufgabe: Roboterschutz	100
3.13	Zusammenfassung	103
3.14	Referenzen	103

4 Lernen im einfachen Netz 105

4.1	Vorgeschichte: Man lässt planen	105
4.2	Lernen im Python-Code	106
4.3	Perceptron-Lernen	107
4.4	Trenngerade für einen Lernschritt	110
4.5	Perceptron-Lernalgorithmus	112
4.6	Die Trenngeraden bzw. Hyperplanes oder auch Hyperebenen für das Beispiel	117

4.7	scikit-learn-kompatibler Estimator	120
4.8	scikit-learn-Perceptron-Estimator	127
4.9	Adaline	129
4.10	Zusammenfassung	139
4.11	Referenzen	141
5	Mehrschichtige neuronale Netze	143
5.1	Ein echtes Problem	143
5.2	XOR kann man lösen	145
5.3	Vorbereitungen für den Start	150
5.4	Der Plan für die Umsetzung	152
5.5	Das Setup (»class«)	153
5.6	Die Initialisierung (»__init__«)	155
5.7	Was für zwischendurch (»print«)	158
5.8	Die Auswertung (»predict«)	159
5.9	Die Verwendung	161
5.10	Zusammenfassung	162
6	Lernen im mehrschichtigen Netz	163
6.1	Wie misst man einen Fehler?	163
6.2	Gradientenabstieg an einem Beispiel	165
6.2.1	Gradientenabstieg – die Idee	165
6.2.2	Algorithmus für den Gradientenabstieg	167
6.3	Ein Netz aus sigmoiden Neuronen	174
6.4	Der coole Algorithmus mit Vorwärts-Delta und Rückwärts-Propagation	176
6.4.1	Die __init__-Methode	176
6.4.2	Die predict-Methode	179
6.4.3	Die fit-Methode	184
6.4.4	Die plot-Methode	186
6.4.5	Alles im Konzert	186

6.5	Ein fit-Durchlauf	189
6.5.1	Initialisierung	191
6.5.2	Forward	192
6.5.3	Output	193
6.5.4	Hidden	194
6.5.5	Delta W_{kj}	195
6.5.6	Delta W_{ji}	197
6.5.7	W_{ji}	197
6.5.8	W_{kj}	197
6.6	Zusammenfassung	198
6.7	Referenz	198
7	Beispiele für tiefe neuronale Netze	199
7.1	Convolutional Neural Networks	199
7.1.1	Die Architektur von Convolutional Networks	201
7.1.2	Der Kodierungsblock	203
7.1.3	Der Prädiktionsblock	209
7.1.4	Trainieren von Convolutional Neural Networks	211
7.2	Transformer Neural Networks	216
7.2.1	Die Netzwerkstruktur	217
7.2.2	Embeddings	219
7.2.3	Positional Encoding	219
7.2.4	Encoder	223
7.2.5	Decoder	224
7.2.6	Trainieren von Transformer Neural Networks	226
7.3	Das Optimierungsverfahren	226
7.3.1	Momentum-Optimierung	227
7.3.2	Adam-Optimierung	228
7.4	Overfitting verhindern	228
7.4.1	Early Stopping	228
7.4.2	Dropout	229
7.5	Zusammenfassung	230
7.6	Referenzen	231

8	Programmierung von Deep Neural Networks mit TensorFlow 2	233
8.1	Convolutional Networks zur Handschriftenerkennung	233
8.1.1	Der Datensatz	233
8.1.2	Ein einfaches CNN	237
8.1.3	Die Ergebnisse	242
8.2	Transfer Learning mit Convolutional Neural Networks	249
8.2.1	Das vortrainierte Netzwerk	251
8.2.2	Datenvorbereitung	253
8.2.3	Das vortrainierte Netz	253
8.2.4	Die Ergebnisse	256
8.3	Transfer Learning mit Transformer Neural Networks	259
8.3.1	Die Transformer-Bibliothek	259
8.3.2	Tokenizer und Modelle	261
8.3.3	Der Model Hub von Hugging Face	262
8.4	Zusammenfassung	264
8.5	Referenzen	265

Teil II Deep Dive

9	Vom Hirn zum Netz	269
9.1	Ihr Gehirn in Aktion	270
9.2	Das Nervensystem	270
9.3	Das Gehirn	271
9.3.1	Die Teile	272
9.3.2	Ein Ausschnitt	273
9.4	Neuronen und Gliazellen	274
9.5	Eine Übertragung im Detail	276
9.6	Darstellung von Zellen und Netzen	279
9.7	Zusammenfassung	280
9.8	Referenzen	281

10	Die Evolution der künstlichen neuronalen Netze	283
10.1	Die 1940er-Jahre	284
10.1.1	1943: McCulloch-Pitts Neurons	284
10.1.2	1949: Donald Hebb	285
10.2	Die 1950er-Jahre	286
10.2.1	1951: Marvin Minsky und Dean Edmonds – SNARC	286
10.2.2	1955/1956: Artificial Intelligence	286
10.2.3	1958: Rosenblatts Perceptron	287
10.2.4	1960: Bernard Widrow und Marcian Hoff – Adaline und Madaline	287
10.3	Die 1960er-Jahre	288
10.3.1	1969: Marvin Minsky und Seymour Papert	288
10.4	Die 1970er-Jahre	288
10.4.1	1972: Kohonen – assoziativer Memory	288
10.4.2	1973: Lighthill Report	288
10.4.3	1974: Backpropagation	289
10.5	Die 1980er-Jahre	289
10.5.1	1980: Fukushimas Neocognitron	289
10.5.2	1982: John Hopfield	291
10.5.3	1982: Kohonens SOM	301
10.5.4	1986: Backpropagation	302
10.5.5	1987: NN-Konferenz	302
10.5.6	1989: Yann LeCun: Convolutional Neural Networks	302
10.6	Die 1990er-Jahre	303
10.6.1	1997 Sepp Hochreiter und Jürgen Schmidhuber – LSTM	303
10.7	Die 2000er-Jahre	303
10.7.1	2006: Geoffrey Hinton et al.	304
10.8	Die 2010er-Jahre	304
10.8.1	2014: Ian J. Goodfellow et al. – Generative Adversarial Networks (GAN)	304
10.8.2	2017: Ashish Vaswani et al. – Attention Is All You Need	306
10.9	Zusammenfassung	307
10.10	Referenzen	307

11	Der Machine-Learning-Prozess	309
11.1	Das CRISP-DM-Modell	309
11.1.1	Geschäfts(prozess)-Verständnis / Business Understanding	310
11.1.2	Datenverständnis / Data Understanding	311
11.1.3	Datenvorbereitung / Data Preparation	311
11.1.4	Modellierung / Modeling	312
11.1.5	Evaluation	312
11.1.6	Einsatz / Deployment	313
11.2	Ethische und rechtliche Aspekte	313
11.2.1	Algorithmische Fairness und Bias	314
11.2.2	Erklärbarkeit und Interpretierbarkeit	317
11.2.3	Ökologische Aspekte	321
11.2.4	Juristische Aspekte	322
11.3	Feature Engineering	324
11.3.1	Feature-Kodierung	326
11.3.2	Feature-Extraktion	337
11.3.3	Der Fluch der Dimensionalität	347
11.3.4	Feature-Transformation	347
11.3.5	Feature-Auswahl	352
11.4	Zusammenfassung	353
11.5	Referenzen	354
12	Lernverfahren	355
12.1	Lernstrategien	355
12.1.1	Überwachtes Lernen (Supervised Learning)	356
12.1.2	Unüberwachtes Lernen (Unsupervised Learning)	360
12.1.3	Verstärkendes Lernen (Reinforcement Learning)	374
12.1.4	Teilüberwachtes Lernen (Semi-supervised Learning)	383
12.2	Werkzeuge	384
12.2.1	Confusion Matrix	385
12.2.2	ROC-Curves	386
12.3	Zusammenfassung	389
12.4	Referenzen	390

13	Anwendungsbereiche und Praxisbeispiele	391
13.1	Warmup	391
13.2	Bildklassifikation	394
13.2.1	Begriffsbestimmung	394
13.2.2	Von Bienen und Hummeln	396
13.2.3	(Vor-)Trainierte Netze	407
13.3	Erträumte Bilder	415
13.3.1	Der Algorithmus	416
13.3.2	Die Implementierung	418
13.4	Deployment mit vortrainierten Netzen	426
13.4.1	Eine Webapplikation für ein neuronales Netz zur Erzeugung von Bildbeschreibungen	426
13.4.2	Eine Webapplikation für die Bilderzeugung	428
13.5	Zusammenfassung	430
13.6	Referenzen	431
Anhang		433
A	Python kompakt	435
B	Mathematik kompakt	465
C	TensorFlow 2 und Keras	485
Index		497