

Roland Schwaiger,
Joachim Steinwendner

Neuronale Netze programmieren mit Python

Inhalt

| | |
|--|-----------|
| Materialien zum Buch | 12 |
| Vorwort | 13 |
| 1 Einleitung | 17 |
| 1.1 Wozu neuronale Netze? | 17 |
| 1.2 Über dieses Buch | 18 |
| 1.3 Der Inhalt kompakt | 19 |
| 1.4 Ist die Biene eine Königin? | 23 |
| 1.5 Ein künstliches neuronales Netz für den Bienenstaat | 24 |
| 1.6 Von der Biologie zum künstlichen Neuron | 28 |
| 1.6.1 Das biologische Neuron und seine technische Kopie | 29 |
| 1.6.2 Das künstliche Neuron und seine Elemente | 30 |
| 1.7 Einordnung und der Rest | 33 |
| 1.7.1 Big Picture | 33 |
| 1.7.2 Artificial Intelligence (künstliche Intelligenz) | 34 |
| 1.7.3 Geschichte | 35 |
| 1.7.4 Machine Learning (maschinelles Lernen) | 36 |
| 1.7.5 Deep Neural Networks | 39 |
| 1.8 Zusammenfassung | 40 |
| 1.9 Referenzen | 41 |

Teil I Up and running

| | |
|---|-----------|
| 2 Das minimale Starterkit für die Entwicklung von neuronalen Netzen mit Python | 45 |
| 2.1 Die technische Entwicklungsumgebung | 45 |
| 2.1.1 Die Anaconda-Distribution | 45 |
| 2.1.2 Unser Cockpit: Jupyter Notebook | 50 |
| 2.1.3 Wichtige Python-Module | 60 |
| 2.2 Zusammenfassung | 62 |

| | | |
|----------|---|-----|
| 3 | Ein einfaches neuronales Netz | 63 |
| 3.1 | Vorgeschichte | 63 |
| 3.2 | Her mit dem neuronalen Netz! | 63 |
| 3.3 | Neuron-Zoom-in | 67 |
| 3.4 | Stufenfunktion | 71 |
| 3.5 | Perceptron | 73 |
| 3.6 | Punkte im Raum – Vektorrepräsentation | 75 |
| 3.6.1 | Aufgabe: Werte vervollständigen | 76 |
| 3.6.2 | Aufgabe: Iris-Datensatz als Scatter-Plot ausgeben | 78 |
| 3.7 | Horizontal und vertikal – Spalten- und Zeilenschreibweise | 81 |
| 3.7.1 | Aufgabe: Ermittlung des Skalarprodukts mit Hilfe von NumPy | 82 |
| 3.8 | Die gewichtete Summe | 84 |
| 3.9 | Schritt für Schritt – Stufenfunktionen | 84 |
| 3.10 | Die gewichtete Summe reloaded | 85 |
| 3.11 | Alles zusammen | 86 |
| 3.12 | Aufgabe – Roboterschutz | 88 |
| 3.13 | Zusammenfassung | 91 |
| 3.14 | Referenzen | 91 |
| | | |
| 4 | Lernen im einfachen Netz | 93 |
| 4.1 | Vorgeschichte: Man lässt planen | 93 |
| 4.2 | Lernen im Python-Code | 95 |
| 4.3 | Perceptron-Lernen | 95 |
| 4.4 | Trenngerade für einen Lernschritt | 99 |
| 4.5 | Perceptron-Lernalgorithmus | 100 |
| 4.6 | Die Trenngeraden bzw. Hyperplanes oder auch Hyperebenen für das Beispiel | 105 |
| 4.7 | scikit-learn-kompatibler Estimator | 109 |
| 4.8 | scikit-learn-Perceptron-Estimator | 116 |
| 4.9 | Adaline | 118 |
| 4.9.1 | Adaline-Lernen | 119 |

| | | |
|----------|--|------------|
| 4.10 | Zusammenfassung | 126 |
| 4.11 | Referenzen | 128 |
| 5 | Mehrschichtige neuronale Netze | 129 |
| 5.1 | Ein echtes Problem | 129 |
| 5.2 | XOR kann man lösen | 131 |
| 5.3 | Vorbereitungen für den Start | 135 |
| 5.4 | Der Plan für die Umsetzung | 137 |
| 5.5 | Das Setup (»class«) | 138 |
| 5.6 | Die Initialisierung (»__init__«) | 140 |
| 5.7 | Was für zwischendurch (»print«) | 142 |
| 5.8 | Die Auswertung (»predict«) | 143 |
| 5.9 | Die Verwendung | 145 |
| 5.10 | Zusammenfassung | 147 |
| 6 | Lernen im mehrschichtigen Netz | 149 |
| 6.1 | Wie misst man einen Fehler? | 149 |
| 6.2 | Gradientenabstieg an einem Beispiel | 151 |
| 6.2.1 | Gradientenabstieg – die Idee | 151 |
| 6.2.2 | Algorithmus für den Gradientenabstieg | 153 |
| 6.3 | Ein Netz aus sigmoiden Neuronen | 160 |
| 6.4 | Der coole Algorithmus mit Vorwärts-Delta und Rückwärts-Multiplikation | 162 |
| 6.4.1 | »__init__«-Methode | 162 |
| 6.4.2 | »predict«-Methode | 165 |
| 6.4.3 | »fit«-Methode | 170 |
| 6.4.4 | »plot«-Methode | 172 |
| 6.4.5 | Alles im Konzert | 173 |
| 6.5 | Ein »fit«-Durchlauf | 175 |
| 6.5.1 | Initialisierung | 177 |
| 6.5.2 | Forward | 178 |
| 6.5.3 | Output | 179 |
| 6.5.4 | Hidden | 180 |

| | | |
|------------|--|-----|
| 6.5.5 | Delta W_{kj} | 182 |
| 6.5.6 | Delta W_{ji} | 183 |
| 6.5.7 | W_{ji} | 183 |
| 6.5.8 | W_{kj} | 183 |
| 6.6 | Zusammenfassung | 184 |
| 6.7 | Referenz | 184 |
| 7 | Convolutional Neural Networks | 185 |
| 7.1 | Aufbau eines CNN | 187 |
| 7.2 | Der Detektionsteil | 188 |
| 7.2.1 | Convolutional Layer | 188 |
| 7.2.2 | Activation Function | 191 |
| 7.2.3 | Pooling Layer | 192 |
| 7.2.4 | Überlappen, ausfüllen und Schrittlänge | 193 |
| 7.3 | Der Identifikationsteil | 195 |
| 7.3.1 | Flatten | 195 |
| 7.3.2 | Softmax | 196 |
| 7.4 | Trainieren von Convolutional Neural Networks | 197 |
| 7.4.1 | Das Problem der explodierenden/verschwindenden Gradienten | 198 |
| 7.4.2 | Das Optimierungsverfahren | 202 |
| 7.4.3 | Verhindern von Overfitting | 204 |
| 7.5 | Zusammenfassung | 206 |
| 7.6 | Referenzen | 206 |
| 8 | Programmierung von Convolutional Neural Networks mit TensorFlow | 207 |
| 8.1 | Einführung in TensorFlow | 207 |
| 8.1.1 | Einrichten der Environments | 208 |
| 8.1.2 | Ein einfaches TensorFlow-Beispiel | 209 |
| 8.2 | Convolutional Networks zur Handschriftenerkennung | 215 |
| 8.2.1 | Der Datensatz | 216 |
| 8.2.2 | Ein einfaches CNN | 219 |
| 8.2.3 | Die Ergebnisse | 224 |

| | | |
|------------|--|-----|
| 8.3 | Transfer Learning mit Convolutional Neural Networks | 228 |
| 8.3.1 | Das vortrainierte Netzwerk | 229 |
| 8.3.2 | Datenvorbereitung | 231 |
| 8.3.3 | Das vortrainierte Netz | 232 |
| 8.3.4 | Die Ergebnisse | 235 |
| 8.4 | Zusammenfassung | 237 |
| 8.5 | Referenzen | 237 |

Teil II Deep Dive

| | | |
|-------------|---|-----|
| 9 | Vom Hirn zum Netz | 241 |
| 9.1 | Ihr Gehirn in Aktion | 241 |
| 9.2 | Das Nervensystem | 242 |
| 9.3 | Das Gehirn | 243 |
| 9.3.1 | Die Teile | 243 |
| 9.3.2 | Ein Ausschnitt | 244 |
| 9.4 | Neuronen und Gliazellen | 245 |
| 9.5 | Eine Übertragung im Detail | 247 |
| 9.6 | Darstellung von Zellen und Netzen | 250 |
| 9.7 | Zusammenfassung | 252 |
| 9.8 | Referenzen | 253 |
| 10 | Die Evolution der neuronalen Netze | 255 |
| 10.1 | 1940er | 255 |
| 10.1.1 | 1943: McCulloch-Pitts Neurons | 256 |
| 10.1.2 | 1949: Donald Hebb | 257 |
| 10.2 | 1950er | 258 |
| 10.2.1 | 1951: Marvin Minsky und Dean Edmonds: SNARC | 258 |
| 10.2.2 | 1956: Artificial Intelligence | 258 |
| 10.2.3 | 1957: Rosenblatts Perceptron | 259 |
| 10.2.4 | 1959: Bernard Widrow und Marcian Hoff: Adaline und Madaline | 259 |
| 10.3 | 1960er | 260 |
| 10.3.1 | 1969: Marvin Minsky und Seymour Papert | 260 |

| | | |
|--------------|---|-----|
| 10.4 | 1970er | 260 |
| 10.4.1 | 1972: Kohonen – assoziativer Memory | 260 |
| 10.4.2 | 1973: Lighthill Report | 260 |
| 10.4.3 | 1974: Backpropagation | 261 |
| 10.5 | 1980er | 261 |
| 10.5.1 | 1980: Fukushimas Neocognitron | 261 |
| 10.5.2 | 1982: John Hopfield | 263 |
| 10.5.3 | 1982: Kohonens SOM | 273 |
| 10.5.4 | 1986: Backpropagation | 274 |
| 10.5.5 | 1987: NN-Konferenz | 274 |
| 10.6 | 1990er | 274 |
| 10.6.1 | 1997 Sepp Hochreiter und Jürgen Schmidhuber – LSTM | 274 |
| 10.7 | 2000er | 275 |
| 10.7.1 | 2006: Geoffrey Hinton et al. | 275 |
| 10.8 | 2010er | 275 |
| 10.8.1 | 2014: Ian J. Goodfellow et al. – Generative Adversarial Networks (GAN) | 275 |
| 10.9 | Zusammenfassung | 277 |
| 10.10 | Referenzen | 278 |
| 11 | Der Machine-Learning-Prozess | 279 |
| 11.1 | Das CRISP-DM-Modell | 279 |
| 11.1.1 | Geschäfts(prozess)-Verständnis | 280 |
| 11.1.2 | Datenverständnis | 281 |
| 11.1.3 | Datenvorbereitung | 281 |
| 11.1.4 | Modellierung | 282 |
| 11.1.5 | Evaluation | 282 |
| 11.1.6 | Einsatz | 283 |
| 11.2 | Feature Engineering | 283 |
| 11.2.1 | Feature-Kodierung | 285 |
| 11.2.2 | Feature-Extraktion | 296 |
| 11.2.3 | Der Fluch der Dimensionalität | 306 |
| 11.2.4 | Feature Transformation | 306 |
| 11.2.5 | Feature-Auswahl | 311 |

| | |
|--|-----|
| 11.3 Zusammenfassung | 312 |
| 11.4 Referenzen | 312 |
| 12 Lernverfahren | 313 |
| 12.1 Lernstrategien | 313 |
| 12.1.1 Überwachtes Lernen (Supervised Learning) | 314 |
| 12.1.2 Unüberwachtes Lernen (Unsupervised Learning) | 318 |
| 12.1.3 Verstärkendes Lernen (Reinforcement Learning) | 332 |
| 12.1.4 Teilüberwachtes Lernen (Semi-supervised Learning) | 349 |
| 12.2 Werkzeuge | 350 |
| 12.2.1 Confusion-Matrix | 350 |
| 12.2.2 ROC-Curves | 352 |
| 12.3 Zusammenfassung | 355 |
| 12.4 Referenzen | 355 |
| 13 Anwendungsbereiche und Praxisbeispiele | 357 |
| 13.1 Warmup | 357 |
| 13.2 Bildklassifikation | 360 |
| 13.2.1 Begriffsbestimmung | 360 |
| 13.2.2 Von Bienen und Hummeln | 362 |
| 13.2.3 (Vor-)Trainierte Netze | 373 |
| 13.3 Erträumte Bilder | 381 |
| 13.3.1 Der Algorithmus | 382 |
| 13.3.2 Die Implementierung | 384 |
| 13.4 Zusammenfassung | 391 |
| 13.5 Referenzen | 392 |
| A Python kompakt | 393 |
| B Mathematik kompakt | 421 |
| Index | 435 |