

1	Einleitung	1
2	Textdaten verarbeiten und vorverarbeiten	7
2.1	Grundlegende Techniken der Verarbeitung von Textdaten	7
2.2	Mit NumPy arbeiten	10
2.3	One-Hot-Encodierung und Bag-of-Words-Modell	15
3	Grundlagen maschinellen Lernens	17
3.1	Lineare Regression	18
3.1.1	Eine Gerade in eine Punktwolke legen	18
3.1.2	Die Lage der Geraden bestimmen	19
3.1.3	Die Qualität eines Modells bestimmen	24
3.1.4	Multivariate Regression	25
3.1.5	Praktische Umsetzung mit Python und Scikit-Learn	26
3.2	Logistische Regression	29
3.2.1	Verfahrensweise	29
3.2.2	Gütemaße	32
3.2.3	Praktische Umsetzung mit Scikit-Learn	33
3.3	Softmax-Regression	36
3.3.1	Verfahrensweise	36
3.3.2	Praktische Umsetzung mit Scikit-Learn	37
4	Einfache Verfahren zur Vektorisierung von Textdaten	41
4.1	One-Hot-Encodierung und Bag-of-Words-Ansatz	42
4.2	N-grams	45
4.3	TF-IDF-Vektorisierung	46
4.4	Umsetzung mit Scikit-Learn	47
4.4.1	Vektorisierung mit dem Count-Vectorizer	47
4.4.2	TF-IDF-Vektorisierung	50
4.4.3	Lemmatisierung	51
4.4.4	Einsatz eines N-gram-Modells	53

5	Deep Learning-Essentials	57
5.1	Neuronen und neuronale Netze	58
5.2	Wie neuronale Netze lernen	61
5.3	Architektur und Einstellungen eines neuronalen Netzes	63
5.3.1	Anzahl der Neuronen in der ersten aktiven Schicht	64
5.3.2	Anzahl der Neuronen in der Ausgabeschicht	64
5.3.3	Aktivierung der Neuronen der Ausgabeschicht	65
5.3.4	Auswahl einer passenden Verlustfunktion	65
5.3.5	Wahl des Optimierers	65
5.3.6	Aktivierung der Neuronen in der verdeckten Schicht	67
5.4	Ein neuronales Netz mit TensorFlow und Keras aufbauen und anlernen	68
5.4.1	Standardisierung der Features	68
5.4.2	Aufbau und Einstellungen eines neuronalen Netzes	71
5.4.3	Anlernen des Modells	75
5.4.4	Steuerung des Anlernprozesses (Early Stopping)	79
5.5	Generalisierung und Überanpassung	83
5.5.1	Regularisierung	88
5.5.2	Dropout	89
5.5.3	Praktische Umsetzung	90
6	Rekurrente Netze	93
6.1	Aufbau und Funktionsweise rekurrenter Netze	94
6.2	Long Short Term Memory (LSTM) und Gated Recurrent Units (GRU)	97
6.3	Praxis rekurrenter Netze: eine automatische Rechtschreibkorrektur	98
6.3.1	Umsetzung der Encodierung	100
6.3.2	Aufbau und Anlernen des rekurrenten Netzes	107
6.3.3	Mit einem bidirektionalen rekurrenten Layer arbeiten	110
6.4	Anlernen neuronaler Netze mit Generatoren	112
6.4.1	Generatoren und Generator-Funktionen in Python	113
6.4.2	Daten batchweise ziehen	114
6.4.3	Neuronale Netze mit Generatoren anlernen	115
6.4.4	Die Rechtschreibkorrektur mit einem Generator anlernen	118
7	Konvolutionale Netze	121
7.1	Funktionsweise konvolutionaler Netze	121
7.2	Sequenzdaten mit konvolutionalen Netzen verarbeiten	124
7.3	Praxis des Anlernens eines konvolutionalen Netzes mit Textdaten	125
8	Word Embedding	129
8.1	Funktionsweise	130
8.2	Aufgabenübergreifende semantische Räume: word2vec- und fastText-Verfahren	133
8.3	Mit Word Embedding-Verfahren in der Praxis arbeiten	135
8.3.1	Vorverarbeitung und Implementierung mit Keras	135

8.3.2	Der Heidegger-Algorithmus: ein generatives Modell zur Erzeugung von Texten	140
8.3.2.1	Aufbau eines generativen Modells	141
8.3.2.2	Vorbereitung der Daten	141
8.3.2.3	Aufbau und Anlernen des Netzes	144
8.3.2.4	Texte erzeugen	146
8.3.2.5	Synonyme Wörter identifizieren	149
8.4	Mit vortrainierten Worteinbettungen arbeiten (fastText)	152
8.4.1	fastText-Vektorräume aufbereiten	153
8.4.2	Austausch der Gewichte eines Embedding Layers	158
8.4.3	Den Vektorraum um unbekannte Wörter erweitern	160
9	Komplexe Lernarchitekturen umsetzen	163
9.1	Die funktionale API von TensorFlow	164
9.2	Ein Modell mit zwei Eingängen aufbauen und anlernen	167
9.2.1	Architektur des Modells	169
9.2.2	Anlernen des Modells	172
10	Sequence-to-Sequence-Modelle	175
10.1	Encoder-Decoder-Modelle mit Teacher Forcing	176
10.2	Attention-Mechanismus	178
10.3	Encoder-Decoder-Architekturen in der Praxis	180
10.3.1	Ein einfaches Encoder-Decoder-Modell	180
10.3.1.1	Vorbereitung der Daten	180
10.3.1.2	Aufbau des Encoder-Decoder-Modells	183
10.3.1.3	Das Inferenzmodell aufbauen und einsetzen	187
10.3.2	Encoder-Decoder-Modelle mit Attention-Mechanismus	191
10.3.2.1	Vorbereitung der Daten	193
10.3.2.2	Zusammenstellung des neuronalen Netzes	195
10.3.2.3	Anlernen des Modells	199
10.3.2.4	Aufbau des Inferenzmodells	199
10.3.2.5	Das Modell für Übersetzungen einsetzen	202
11	Transformers	207
11.1	Aufbau und Funktionsweise	209
11.1.1	Self-Attention	209
11.1.2	Die Transformer-Architektur	211
11.2	Subwort-Tokenisierung	214
11.3	Mit der Hugging Face-Bibliothek arbeiten	215
11.3.1	Hauptklassen der Transformers-Bibliothek	216
11.3.2	Mit der Hugging Face-Pipeline arbeiten	217
11.3.3	Mit der Tokenizer-Klasse arbeiten	220
11.3.4	Mit der Model-Klasse arbeiten	224

11.3.5	Fine Tuning vortrainierter Netze	225
11.3.5.1	Ein vortrainiertes Modell mit einem nichttrainierten Kopf laden	225
11.3.5.2	Eine Durchleitung organisieren	227
11.3.5.3	Teile des Netzes auf nichttrainierbar stellen	228
11.3.5.4	Das Modell anlernen	230
12	Diskussion und Ausblick	233
	Literaturverzeichnis	239
	Stichwortverzeichnis	243