

Vorwort	XI
1 Einführung	1
Das Paradigma des überwachten Lernens	2
Beobachtung und Zielcodierung	5
1-aus-n-Darstellung	6
TF-Darstellung	7
TF-IDF-Darstellung	8
Zielcodierung	9
Berechnungsgraphen	10
Grundlagen von PyTorch	11
PyTorch installieren	13
Tensoren erstellen	13
Typ und Größe von Tensoren	16
Tensor-Operationen	17
Indizieren, Slicing und Verknüpfen	20
Tensoren und Berechnungsgraphen	23
CUDA-Tensoren	25
Übungen	27
Lösungen	27
Zusammenfassung	28
Literaturhinweise	28
2 Kurzer Abriss des traditionellen NLP	29
Korpora, Token und Typen	29
Monogramme, Bigramme, Trigramme, ..., N-Gramme	32
Lemmas und Stämme	33
Sätze und Dokumente kategorisieren	34
Wörter kategorisieren: POS-Tagging	34
Spannen kategorisieren: Phrasenerkennung und Eigennamenerkennung	35
Struktur von Sätzen	36

Wortbedeutungen und Semantik	37
Zusammenfassung	38
Literaturhinweise	38
3 Grundlegende Komponenten von neuronalen Netzen	39
Das Perzeptron: Das einfachste neuronale Netz	39
Aktivierungsfunktionen	41
Sigmoid	41
Tanh.	42
ReLU	43
Softmax	44
Verlustfunktionen	45
Mittlere quadratische Abweichung	46
Kategorischer Kreuzentropie-Verlust	46
Binärer Kreuzentropie-Verlust	48
Überwachtes Training unter der Lupe	49
Die Spielzeugdaten konstruieren	49
Gradientenbasiertes überwachtes Lernen	52
Unterstützende Trainingskonzepte	53
Modellperformance richtig messen: Bewertungskennzahlen	53
Modellperformance richtig messen: das Dataset aufteilen	54
Feststellen, wann das Training beendet werden sollte	55
Die richtigen Hyperparameter finden	55
Regularisierung	56
Beispiel: Stimmungen von Restaurantbewertungen klassifizieren	57
Das Yelp-Dataset für Bewertungen	58
Die Dataset-Darstellung von PyTorch verstehen	60
Vocabulary, Vectorizer und DataLoader	62
Ein Perzeptron-Klassifizierer	68
Die Trainingsroutine	69
Bewertung, Inferenz und Inspektion	75
Zusammenfassung	78
Literaturhinweise	79
4 Feedforward-Netze für NLP	81
Das Multilayer-Perzeptron	82
Ein einfaches Beispiel: XOR	84
MLPs in PyTorch implementieren	85
Beispiel: Nachnamen mit einem MLP klassifizieren	89
Das Nachnamen-Dataset	91
Vocabulary, Vectorizer und DataLoader	92
Das SurnameClassifier-Modell	94

Die Trainingsroutine.	95
Modellauswertung und Vorhersage	97
MLPs regularisieren: Gewichtsregularisierung und strukturelle Regularisierung (oder Dropout)	99
CNNs.	101
CNN-Hyperparameter	102
CNNs in PyTorch implementieren	107
Beispiel: Nachnamen mit einem CNN klassifizieren.	110
Die Klasse SurnameDataset	111
Vocabulary, Vectorizer und DataLoader	112
Den SurnameClassifier mit CNNs neu implementieren.	113
Die Trainingsroutine.	114
Modellbewertung und Vorhersage	115
Verschiedene Themen in CNNs	116
Pooling	116
Batch-Normalisierung (BatchNorm)	117
Netzwerk-in-Netzwerk-Verbindungen (1x1-Faltungen).	118
Residual-Verbindungen/Residual-Block.	118
Zusammenfassung.	119
Literaturhinweise.	120
5 Wörter und Typen einbetten.	121
Warum Einbettungen lernen?	122
Effizienz von Einbettungen.	123
Ansätze für das Lernen von Worteinbettungen	124
Die praktische Verwendung von vortrainierten Worteinbettungen.	125
Beispiel: Erlernen der CBOW-Einbettungen.	131
Das Frankenstein-Dataset.	132
Vocabulary, Vectorizer und DataLoader	134
Das CBOWClassifier-Modell	134
Die Trainingsroutine.	136
Modellbewertung und Vorhersage	136
Beispiel: Transfer-Lernen mit vortrainierten Einbettungen für Dokumentklassifizierung.	137
Das Dataset AG News	138
Vocabulary, Vectorizer und DataLoader	139
Das NewsClassifier-Modell.	141
Die Trainingsroutine.	144
Modellbewertung und Vorhersage	145
Zusammenfassung.	146
Literaturhinweise.	147

6	Sequenzmodellierung für NLP	149
	Einführung in rekurrente neuronale Netze	150
	Ein Elman-Netz implementieren	153
	Beispiel: Die Nationalität von Nachnamen mit einem Zeichen-RNN	
	klassifizieren	155
	Die Klasse SurnameDataset	155
	Die Datenstrukturen der Vektorisierung	156
	Das SurnameClassifier-Modell	157
	Die Trainingsroutine und die Ergebnisse	160
	Zusammenfassung	161
	Literaturhinweise	161
7	Intermediäre Sequenzmodellierung für NLP	163
	Das Problem mit einfachen RNNs (oder Elman-Netzen)	164
	Gating als eine Lösung für Herausforderungen von einfachen	
	RNNs	165
	Beispiel: Nachnamen mit Zeichen-RNN generieren	167
	Die Klasse SurnameDataset	167
	Die Vektorisierungs-Datenstrukturen	168
	Vom ElmanRNN zur GRU	170
	Modell 1: Das unkonditionierte SurnameGenerationModel	171
	Modell 2: Das konditionierte SurnameGenerationModel	172
	Die Trainingsroutine und die Ergebnisse	173
	Tipps und Tricks für das Training von Sequenzmodellen	179
	Literaturhinweise	180
8	Erweiterte Sequenzmodellierung für NLP	181
	Sequenz-zu-Sequenz-Modelle, Encoder-Decoder-Modelle und	
	konditionierte Generierung	181
	Mehr von einer Sequenz erfassen: Bidirektionale rekurrente	
	Modelle	184
	Mehr von einer Sequenz erfassen: Attention	186
	Attention in tiefen neuronalen Netzen	188
	Sequenzgenerierungsmodelle bewerten	190
	Beispiel: Neuronale maschinelle Übersetzung	193
	Das Dataset für maschinelle Übersetzung	193
	Eine Vektorisierungs-Pipeline für NMT	194
	Im NMT-Modell codieren und decodieren	198
	Die Trainingsroutine und die Ergebnisse	208
	Zusammenfassung	211
	Literaturhinweise	211

9	Klassiker, Grenzen und nächste Schritte	213
	Was haben Sie bisher gelernt?	213
	Zeitlose Themen in NLP	214
	Dialog- und interaktive Systeme	214
	Diskurs	215
	Informationsextraktion und Text Mining	216
	Analyse und Abrufen von Dokumenten	217
	Grenzen in NLP	217
	Entwurfsmuster für NLP-Produktionssysteme	219
	Wie geht es weiter?	223
	Literaturhinweise	224
	Index	227