

Index

- activation functions
 - hyperbolic tangent, 94
 - Leaky ReLU, 96
 - logistic, 30
 - rectified linear unit, 94
 - ReLU, 94
 - sigmoid, 30
 - softmax, 41, 76
 - tanh, 94
- applications
 - dependency parsing, 255
 - Universal dependency types, 255
 - machine translation, 229, 269
 - fine-tuning, 237
 - greedy decoding, 235
 - named entity recognition, 252
 - annotation schemas, 253
 - part-of-speech tagging, 147
 - recurrent neural networks, 171, 249
 - transformer networks, 204, 249
 - Universal tags, 249
 - question answering, 264
 - extractive, 265
 - multiple-choice, 267
 - relation extraction, 260
 - text classification, 11
 - bag of words, 12, 109
 - distributional representations, 140, 246
 - recurrent neural networks, 247
 - transformer networks, 196, 248
- ASCII character encoding, 301
- ASCII control characters, 301
- ASCII printable characters, 301
- avoiding overflow, 58
- backpropagation
 - equations, 79
- binary_classification_report, 56
- bit, 301
- byte, 302
- character encodings, 301
- chardet, 305
- classification task
 - binary, 10
 - multiclass, 10
- classifier, 9
- common rules of computation for derivatives, 40
- CoNLL-U format, 165
- cosine similarity, 18
- cost functions
 - binary cross-entropy, 43, 98
 - cross-entropy, 43, 99
 - mean squared error, 80, 97
 - negative log likelihood, 34
 - regularization, 99
- CountVectorizer, 52
 - fit, 52
 - fit_transform, 52
 - transform, 52
- curse of dimensionality, 117
- dataset partitions
 - development, 11
 - testing, 11
 - training, 11
- datasets
 - AG News, 62, 108
 - AnCora, 165
 - Large Movie Review Dataset, 50
 - WMT 2016, 229
- deep learning, 1
- distributional hypothesis, 117
- distributional representations
 - co-occurrence vectors, 118
 - low-rank approximations, 120

322 Index

- pretrained word embeddings, 132
 - singular value decomposition, 121
 - word analogies, 135, 139
 - word similarity, 134, 137
 - word2vec, 123
 - training algorithm, 127
- document-term matrix, 52
- dot product, 15
- dynamic programming, 161
- error-driven learning, 18
- evaluation measures
 - accuracy, 12
 - binary F1, 13
 - binary precision, 13
 - binary recall, 13
 - BLEU, 217
 - class precision, 44
 - class recall, 44
 - macro F1, 45
 - macro precision, 45
 - macro recall, 45
 - micro F1, 45
 - micro precision, 45
 - micro recall, 45
- features, 8
 - feature matrix, 10
- feed-forward neural networks
 - architecture, 73
 - input layer, 75
 - intermediate layers, 75
 - output layer, 76
 - PyTorch implementation, 109
- Gensim, 133
 - KeyedVectors, 133
 - add_vectors, 142
 - get_normed_vectors, 136
 - index_to_key, 137
 - key_to_index, 137
 - most_similar, 134
 - loading embedding weights, 133
 - word analogies, 135
- GloVe embedding weights, 132
- gradient descent, 34
 - AdaDelta, 93
 - AdaGrad, 93
 - Adam, 94
 - adaptive learning rate, 93
 - batch, 88
 - minibatch, 89
 - momentum, 91
 - Nesterov momentum, 91
 - RMSProp, 93
 - stochastic, 38, 78, 87
- Hugging Face
 - </s> token, 196
 - <s> token, 196
 - [CLS] token, 195
 - [SEP] token, 195
 - aligning word labels with sub-words, 205
 - AutoConfig, 201
 - from_pretrained, 201
 - AutoModelForSeq2SeqLM, 230
 - from_pretrained, 244
 - AutoTokenizer, 195, 230
 - from_pretrained, 195, 244
 - BertPreTrainedModel, 198
 - from_pretrained, 198, 201
 - init_weights, 198
 - BLEU metric (sacrebleu), 234
 - DataCollatorForSeq2Seq, 239
 - DataCollatorForTokenClassification, 210
 - Dataset, 196, 204
 - map, 197
 - to_pandas, 206
 - DatasetDict, 196, 204
 - EvalPredictions, 202
 - ignoring sub-words during training, 205
 - implementing custom encoder model for
 - sequence classification, 198
 - implementing custom encoder model for
 - token classification, 208
 - implementing custom encoder-decoder
 - model for machine translation, 237
 - load_dataset, 230
 - Metric, 234
 - preserving word boundaries during
 - tokenization, 205
 - RobertaPreTrainedModel, 208
 - from_pretrained, 209
 - Seq2SeqTrainer, 241
 - Seq2SeqTrainingArguments, 241
 - T5, 229
 - Trainer, 201
 - compute_metrics, 202, 210
 - create_model_card, 243
 - evaluate, 243
 - generate, 241
 - predict, 203, 211
 - save_metrics, 243
 - save_model, 242
 - save_state, 243
 - train, 203, 210, 242
 - TrainingArguments, 201
 - using a pretrained encoder-decoder model, 231
 - using checkpoints, 242
 - XLNetForTokenClassification, 206
- ISO-8859-1 character encoding, 302

- JIS character encoding, 302
- labels, 9
 - label vector, 10
- Latin-1 character encoding, 302
- logistic regression
 - binary cost function, 38
 - decision function, 30
 - drawbacks, 46
 - multiclass cost function, 42
 - NumPy implementation, 57
 - PyTorch implementation, 60, 69
 - training algorithm, 31, 38, 43
- logit, 77
- matplotlib
 - plot, 113, 212
- model, 16
- most_similar_words, 137
- natural language processing, 1
- NLTK
 - word_tokenize, 67
- nonlinear classifier, 23
- NumPy
 - arange, 287
 - argmax, 295
 - argsort, 138, 295
 - array, 287
 - dtype, 289
 - min, 296
 - shape, 288
 - at operator, 291
 - broadcasting, 294
 - column_stack, 58
 - dot product, 55
 - expand_dims, 294
 - finfo, 58
 - indexing, 292
 - isin, 139
 - linspace, 288
 - ones, 58, 288
 - random, 58
 - reshape, 211
 - vectorized operations, 290
 - zeros, 288
- optimizers
 - AdaDelta, 93
 - AdaGrad, 93
 - Adam, 94
 - RMSProp, 93
- overfitting, 11, 85
- pandas
 - insert, 63
- read_csv, 63
- Series, 67
 - explode, 67
 - map, 67
 - progress_map, 67
 - to_numpy, 69
 - unique, 169
 - value_counts, 67
- parameter initialization, 103
 - Glorot, 104
 - Xavier, 104
- parameter normalization, 104
 - batch normalization, 104
 - layer normalization, 104
- perceptron
 - average perceptron, 24
 - bias term, 21
 - decision boundary and convergence, 20
 - decision function, 16
 - definition, 14
 - drawbacks, 26
 - NumPy implementation, 53
 - training algorithm, 17
 - voting perceptron, 22
- Python
 - built-in functions, 282
 - bytes, 304
 - class inheritance, 285
 - classes and objects, 284
 - containers, 273
 - context managers, 286
 - decode, 304
 - dict, 277
 - del, 278
 - double star operator, 281
 - in, 277
 - items, 277
 - keys, 277
 - values, 277
 - dunder methods, 284
 - encode, 304
 - enumerate, 283
 - f-strings, 278
 - Format Specification Mini-Language, 279
 - functions, 279
 - assigning parameters by name, 281
 - default parameter values, 282
 - lambda, 281
 - list, 273
 - append, 273
 - insert, 273
 - mappings, 277
 - other tutorials, 272
 - range, 282
 - sequences, 273
 - concatenation, 274
 - in, 276

324 Index

- indexes, 275
 - len, 274
 - negative indexes, 275
 - slices, 276
 - star operator, 274
- star operator, 281
- str, 274
- string formatting, 278
- tuple, 273
- unicodedata, 306
- unpacking dictionaries, 281
- unpacking sequences, 281
- with, 286
- zip, 283
- PyTorch
 - Adam, 111
 - BCEWithLogitsLoss, 60
 - collate_fn, 170
 - cpu, 112
 - CrossEntropyLoss, 71
 - cuda
 - is_available, 299
 - DataLoader, 108, 173
 - Dataset, 108
 - detach, 112
 - device, 300
 - Dropout, 110
 - Embedding, 144
 - from_pretrained, 144
 - flatten, 174
 - from_numpy, 297
 - GPU usage, 299
 - implementing custom dataset, 109, 143, 169
 - implementing custom module, 110, 144, 171
 - Linear, 60, 107
 - LSTM, 171
 - Module, 110, 298
 - __init__, 298
 - eval, 112
 - forward, 298
 - to, 111, 300
 - train, 112
 - no_grad, 112
 - numpy, 112
 - pack_padded_sequence, 171
 - PackedSequence, 171
 - pad_packed_sequence, 171
 - pad_sequence, 170
 - ReLU, 110
 - Sequential, 110
 - SGD, 60
 - tensor, 69, 297
 - numpy, 297
 - to, 300
 - tolist, 297
 - view, 174
- recurrent neural networks
 - acceptor, 149
 - bidirectional, 150
 - conditional random fields, 155
 - forward algorithm, 156
 - Viterbi algorithm, 161
 - deep, 150
 - encoder-decoder, 150, 216, 219
 - encoder-decoder with attention, 221
 - long short-term memory, 152
 - LSTM, 152
 - stacked, 150
 - transducer, 149
 - vanilla, 148
- regression task, 10
- regularization
 - dropout, 101
 - L1, 100
 - L2, 100
- scikit-learn
 - classification_report, 71, 203, 211
 - confusion_matrix, 212
 - train_test_split, 108
- Spanish GloVe embeddings, 166
- temporal averaging, 102
- tqdm, 55
- transformer networks
 - add and normalize, 186
 - byte pair encoding, 186
 - contextualized embeddings, 178
 - encoder-decoder, 224
 - implementation, 229, 237
 - fine-tuning, 190
 - heads, 185
 - layer architecture, 179
 - masked language model, 188
 - next sentence prediction, 189
 - positional embeddings, 181
 - pretraining, 188
 - self-attention, 182
 - tokenization, 186
 - implementation, 194
- Unicode case folding, 307
- Unicode character names, 305
- Unicode code point, 303
- Unicode combining characters, 303
- Unicode normalization
 - canonical equivalence, 306
 - compatibility equivalence, 306
 - composition, 306
 - decomposition, 306
 - Normalization Form C (NFC), 306
- Unicode numeric values, 304

Index	325
Unicode standard, 302	
UTF-8, 303	
vanishing gradient, 84	
recurrent neural networks, 151	
Vauquois triangle, 216	
Windows-1252 character encoding, 302	
word analogy implementation, 139	
word frequencies, 11	
Zipf’s law, 117	