

index

Symbols

[BOS] (beginning of sequence) token 32
[EOS] (end of sequence) token 32
[PAD] (padding) token 32
@ operator 261
%timeit command 282
<|endoftext|> token 34
<|unk|> tokens 29–31, 34
== comparison operator 277

Numerics

04_preference-tuning-with-dpo folder 247
124M parameter 161
355M parameter 227

A

AdamW optimizer 148, 294
AI (artificial intelligence) 252
allowed_max_length 224, 233, 309
Alpaca dataset 233, 296
alpha scaling factor 328
architectures, transformer 7–10
argmax function 134, 152–155, 190, 277
arXiv 248
assign utility function 165
attention mechanisms
 causal 74–82
 coding 50, 54
 implementing self-attention with trainable
 weights 64–74
 multi-head attention 82–91

problem with modeling long sequences 52
self-attention mechanism 55–64
attention scores 57
attention weights, computing step by step 65–70
attn_scores 71
autograd engine 264
automatic differentiation 263–265
 engine 252
 partial derivatives and gradients 263
autoregressive model 13
Axolotl 249

B

backpropagation 137
.backward() method 112, 318
Bahdanau attention mechanism 54
base model 7
batch normalization layers 276
batch_size 233
BERT (bidirectional encoder representations from
transformers) 8
BPE (byte pair encoding) 32–35

C

calc_accuracy_loader function 192
calc_loss_batch function 145, 193–194
calc_loss_loader function 144, 194
calculating, training and validation 140, 142
CausalAttention class 80–81, 86, 90
 module 83–84
 object 86

causal attention mask 190
 causal attention mechanism 74–82
 cfg dictionary 115, 119
 classification
 fine-tuning
 categories of 170
 preparing dataset 172–175
 fine-tuning for
 adding classification head 183–190
 calculating classification loss and accuracy 190–194
 supervised data 195–200
 using LLM as spam classifier 200
 tasks 7
 classify_review function 200
 clip_grad_norm_ function 317
 clipping, gradient 317
 code for data loaders 301
 coding
 attention mechanisms 54
 GPT model 117–122
 collate function 211
 computation graphs 261
 compute_accuracy function 277–278
 computing gradients 258
 connections, shortcut 109–113
 context, adding special tokens 29–32
 context_length 47, 95
 context vectors 57, 64, 85
 conversational performance 236
 converting tokens into token IDs 24–29
 cosine decay 313, 316
 create_dataloader_v1 function 39
 cross_entropy function 138–139
 CUDA_VISIBLE_DEVICES environment variable 286
 custom_collate_draft_1 215
 custom_collate_draft_2 218
 custom_collate_fn function 224, 308

D

data, sampling with sliding window 35–41
 DataFrame 173
 data list 207, 209
 DataLoader class 38, 211, 224, 270–272
 data loaders 175–181
 code for 301
 creating for instruction dataset 224–226
 efficient 270–274
 Dataset class 38, 177, 270–272, 274

datasets
 downloading 207
 preparing 324
 utilizing large 10
 DDP (DistributedDataParallel) strategy 282
 ddp_setup function 286
 decode method 27, 33–34
 decoder 52
 decoding strategies to control randomness 151–159
 modifying text generation function 157
 temperature scaling 152–155
 top-k sampling 155
 deep learning 253
 library 252
 destroy_process_group function 284
 device variable 224
 dim parameter 101–102
 DistributedDataParallel class 284
 DistributedSampler 283–284
Dolma: An Open Corpus of Three Trillion Tokens for LLM Pretraining Research (Soldaini et al.) 11
 dot products 58
 d_out argument 90, 301
 download_and_load_gpt2 function 161, 163, 182
 drop_last parameter 273
 dropout
 defined 78
 layers 276
 drop_rate 95
 .dtype attribute 259
 DummyGPTClass 98
 DummyGPTModel 95, 97–98, 117
 DummyLayerNorm 97, 99, 117
 placeholder 100
 DummyTransformerBlock 97, 117

E

emb_dim 95
 Embedding layer 161
 embedding size 46
 emergent behavior 14
 encode method 27, 33, 37
 encoder 52
 encoding word positions 43–47
 entry dictionary 209
 eps variable 103
 .eval() mode 126
 eval_iter value 200
 evaluate_model function 147–148, 196

F

feedforward layer 267
 FeedForward module 107–108, 113
 feed forward network, implementing with GELU activations 105–109
 find_highest_gradient function 318
 fine-tuning
 categories of 170
 creating data loaders for instruction dataset 224–226
 evaluating fine-tuned LLMs 238–247
 extracting and saving responses 233–238
 for classification 169
 adding classification head 183–190
 calculating classification loss and accuracy 190–194
 data loaders 175–181
 fine-tuning model on supervised data 195–200
 initializing model with pretrained weights 181
 preparing dataset 172–175
 using LLM as spam classifier 200
 instruction data 230–233
 instruction fine-tuning, overview 205
 LLMs, to follow instructions 204
 organizing data into training batches 211–223
 supervised instruction fine-tuning, preparing dataset for 207–211
 FineWeb Dataset 295
 first_batch variable 39
 format_input function 209–210, 242, 307
 forward method 97, 109, 267, 330
 foundation model 7
 fully connected layer 267
 functools standard library 224

G

GELU (Gaussian error linear unit) 105, 107, 293
 activation function 104, 111
 GenAI (generative AI) 3
 generate_and_print_sample function 147–148, 151, 154
 generate function 157, 159, 167, 228, 234–235, 237, 305
 generate_model_scores function 246
 generate_simple function 157, 159
 generate_text_simple function 125–126, 131–132, 134, 148, 151–153
 generative text models, evaluating 129

__getitem__ method 271
 Google Colab 257
 GPT-2 94
 model 230
 tokenizer 176
 gpt2-medium355M-sft.pth file 238
 GPT-3 11, 94
 GPT-4 239
 GPT_CONFIG_124M dictionary 95, 97, 107, 116–117, 120, 127, 130
 GPTDatasetV1 class 38–39
 gpt_download.py Python module 161
 GPT (Generative Pre-trained Transformer) 8, 18, 93
 architecture 12–14
 coding 117–122
 coding architecture 93–99
 implementing feed forward network with GELU activations 105–109
 implementing from scratch, shortcut connections 109–113
 implementing from scratch to generate text 92, 122
 implementing model from scratch 99–105, 113–116
 GPTModel 119, 121–122, 133, 146, 182, 330
 class 122, 130, 182, 326
 code 141
 implementation 166
 instance 131, 159, 164–167
 GPUs (graphics processing units), optimizing training performance with 279–288
 .grad attribute 318
 grad_fn value 268
 grad function 264
 gradient clipping 313, 317
 gradients 263
 greedy decoding 125, 152

I

information leakage 76
 __init__ constructor 71, 81, 119, 266–267, 271
 initializing model 326
 initial_lr 314
 init_process_group function 284
 input_chunk tensor 38
 input_embeddings 47
 'input' object 208
 instruction data, fine-tuning LLMs on 230–233
 instruction dataset 205
 InstructionDataset class 212, 224, 308