



Building Large Language Models from Scratch

Design, Train, and Deploy LLMs
with PyTorch

—
Dilyan Grigorov

Apress®

Building Large Language Models from Scratch

**Design, Train, and Deploy LLMs
with PyTorch**

Dilyan Grigorov

Apress®

Building Large Language Models from Scratch: Design, Train, and Deploy LLMs with PyTorch

Dilyan Grigorov
Floor 7,
Varna, Bulgaria

ISBN-13 (pbk): 979-8-8688-2296-4

<https://doi.org/10.1007/979-8-8688-2297-1>

ISBN-13 (electronic): 979-8-8688-2297-1

Copyright © 2026 by Dilyan Grigorov

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Celestin Suresh John

Coordinating Editor: Gryffin Winkler

Cover image designed by Giovanni_cg on unsplash.com

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a Delaware LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub. For more detailed information, please visit <https://www.apress.com/gp/services/source-code>.

If disposing of this product, please recycle the paper

*To the **University of Bath**, where ideas flourish and boundaries are meant to be pushed. Within these walls, I found not just a place to study but a home where curiosity is celebrated and ambitious dreams are nurtured into reality.*

*To **Dr. Harish Tayyar Madabushi**, my supervisor, mentor, and guide through the intricate world of language models. Your unwavering belief in my potential gave me the courage to tackle challenges I once thought insurmountable.*

*To my son and my family!
With profound gratitude and warmest regards.*

Table of Contents

About the Authorxxi

About the Technical Reviewerxxiii

Introductionxxv

Chapter 1: What Is a Large Language Model? Getting Started with Libraries and Environment Setup for Building an LLM from Scratch 1

 Getting Started with the Foundations of Language Modeling..... 1

 What Is a Large Language Model (LLM)? 3

 The Attention Mechanism—Cornerstone of Modern Large Language Models (LLMs) 6

 The Tools We Will Use to Build a Large Language Model from Scratch..... 12

 Why Python for Building Large Language Models? 14

 Why Jupyter Notebook for Building Large Language Models?..... 16

 Why PyTorch for Building Large Language Models? 18

 Why CUDA for Building Large Language Models? 20

 Why NumPy and Matplotlib for Building Large Language Models?..... 29

 TinyStories, Shakespeare, *The Wizard of Oz*, and OpenWebText 38

 Why PyLZMA for Building Large Language Models?..... 41

 Why PyLZMA Matters for LLM Development 41

 Advanced Concepts 44

 In Summary 46

Chapter 2: Foundational Concepts in LLM Development 47

 Large Language Models Common Architecture 47

 Transformer Architecture: Conceptual and Mathematical Deep Dive 48

 Input Embedding 49

 Positional Encoding 50

 Attention Mechanism..... 52

TABLE OF CONTENTS

- Output Layer 56
- Loss Function 57
- Encoder-Decoder Interaction..... 57
- Complexity Analysis..... 58
- Advantages of the Transformer 58
- Limitations..... 59
- Applications..... 60
- Variants and Improvements..... 60
- The Comprehensive Mathematics Behind Training Large Language Models..... 61
 - Explaining the Purpose and Scope 61
 - Defining the Goal of Language Modeling..... 61
 - Modeling Sequential Probability 62
 - Defining the Loss Function 62
 - Softmax for Probability Distribution 62
 - Practical Considerations..... 63
 - Masked Language Modeling..... 63
 - Optimization: Gradient-Based Learning..... 64
 - Gradient Descent 64
 - Stochastic Gradient Descent 64
 - Adam Optimizer 65
 - Why Adam Works for LLMs 65
 - Backpropagation 66
 - Role of Automatic Differentiation..... 66
 - Regularization Techniques..... 66
 - Why Dropout Helps 67
 - Penalizing Large Weights 67
 - Implementation in Adam..... 67
 - Label Smoothing..... 68
 - Impact on Training 68
 - Fine-Tuning and RLHF 68
 - Supervised Fine-Tuning (SFT)..... 68

Differences from Pretraining	69
Reinforcement Learning from Human Feedback (RLHF).....	69
Aligning with Human Preferences	69
Why RLHF Is Effective.....	70
Summarizing the Mathematical Framework.....	70
Modern LLM Architecture Characteristics As of the End of 2024 and During 2025	70
In Summary.....	73
Chapter 3: Building a Tokenizer for the Transformers Architecture Model	75
What Is Tokenization?	76
Why Is Tokenization Important?	77
Types of Tokenizers	78
Word-Based Tokenizers	78
Character-Based Tokenizers	78
Subword-Based Tokenizers	79
Rule-Based Tokenizers	79
Byte-Pair Encoding (BPE): A Deep Dive	80
Origins of BPE.....	80
BPE Algorithm Outline and Mathematical Formulation	81
Identify Frequent Pairs	81
Replace and Record	81
Repeat Until No Gains	82
Decompression (Decoding).....	82
BPE Algorithm Example.....	82
Concrete Example of the Encoding Part	82
Encoding New Text	86
Concrete Example of the Decoding Part.....	86
Optimization Objective.....	87
Vocabulary Size and Hyperparameters	87
Step-by-Step Process of Building a BPE Tokenizer.....	88
Initialize the Vocabulary.....	88
Count Pair Frequencies	88

TABLE OF CONTENTS

- Merge the Most Frequent Pair 89
- Repeat Merging 89
- Tokenize New Text 89
- Map Tokens to IDs 90
- Applications of BPE 90
- Advantages of BPE 90
- Limitations of BPE 91
- Comparison with Other Tokenization Methods 91
- Practical Considerations 92
- BPE Implementation Walkthrough 93
 - Train Method Code Breakdown—Step by Step 98
 - The Encode Method 102
 - The Decode Method: One of the Most Important Ones 103
 - Why This Method Is One of the Most Important 107
 - Save and Load Methods 108
- Helping Functions 110
- Testing Our Tokenizer for Accuracy 117
 - Tokenizer Output 123
- Analysis of the Output 127
 - Training the Tokenizer 127
 - Learned Merges 128
 - Encoding and Decoding Examples 129
 - Step-by-Step BPE Trace 130
 - Saving and Loading 131
- Chapter 4: RMS Normalization and Model Configuration 133**
 - Model Parameters Configuration and Mathematical Foundations 134
 - Mathematical Notation 135
 - Global Tensor Shapes and Consistency 135
 - num_hidden_layers—The Depth of Abstraction 136
 - vocab_size—The Universe of Symbols 136

hidden_size—The Resolution of Thought.....	137
intermediate_size—The Breathing Room of the Network.....	138
head_dim—The Grain of Attention	139
num_attention_heads—The Many Eyes of the Model.....	139
num_key_value_heads—Sharing the Burden.....	140
sliding_window—Constraining Attention	141
initial_context_length—How Far the Model Can See.....	142
The Geometry of Position—rope_theta, rope_scaling_factor, rope_ntk_alpha, rope_ntk_beta	142
rope_theta—The Base Frequency	143
rope_scaling_factor—Stretching the Map.....	143
rope_ntk_alpha—Balancing Long-Context Stability	143
rope_ntk_beta—Controlling the Rate of Growth.....	144
swiglu_limit—Taming the Activations	144
What Is SwiGLU?	144
Background on Related Concepts	145
How SwiGLU Works	145
Advantages and Usage	145
SwiGLU Enhances FFN Expressiveness in GPT Architectures with Gated Activations.....	146
RMS Normalization in GPT Architectures	147
What Is RMS Normalization?	147
Why Is RMSNorm Used in GPT Architectures?.....	148
How Is RMSNorm Used in GPT Architectures?	149
RMS Normalization for Our Custom Large Language Model.....	150
Comprehensive Explanation of RMSNorm Code	150
Detailed Code Explanation.....	151
Mathematical Summary	157
Implementation Details and Design Choices	157
Summary—RMS Normalization and Model Configuration.....	158

Chapter 5: Rotary Positional Embeddings: Integrating NTK and YaRN Scaling 161

- Rotary Positional Embeddings: An In-Depth and Comprehensive Exploration 161
 - The Fundamental Need for Positional Embeddings in Transformers..... 163
 - Historical Evolution of Positional Encodings..... 163
 - Pretransformer Approaches: Sequential Processing 163
 - Early Transformer Positional Encodings 164
 - Learned Absolute Positional Embeddings 166
 - Relative Positional Encodings..... 166
 - Emergence of Rotary Positional Embeddings (RoPE) 168
 - Summary of Traditional Positional Embedding Approaches 169
 - Absolute vs. Relative Positional Embeddings 171
 - Mathematical Formulation of RoPE 171
 - 2D Intuition and Derivation 171
 - Advantages of RoPE 178
 - Applications in Modern Models 179
 - Variants and Extensions 180
 - Why Rotary Positional Embeddings Are Essential to Large Language Models 180
 - RoPE Embeddings Implementation 182
 - Step-by-Step Explanation..... 183
 - Key Features and Safety Considerations..... 186
 - Example Workflow 186
 - Geometric Interpretation 187
- RotaryEmbedding Class Definition and Methods 187
 - Class Structure 190
 - Detailed Explanation..... 191
 - Key Features and Safety Considerations..... 197
 - Example Workflow 198
 - Geometric Interpretation 198
- What Is NTK (Neural Tangent Kernel) in the Context of RoPE..... 199
 - Implementation in RotaryEmbedding 200
- What Is YaRN (Yet Another RoPE Extension)..... 201

YaRN's Scaling Approach.....	202
Implementation in RotaryEmbedding	203
Summary	204
Chapter 6: Scaled Dot-Product Attention Core—Sliding Window and Grouped Query Attention—The Core Behind All Transformer Models.....	207
What Is Scaled Dot-Product Attention (SDPA)?	207
Historical Evolution and Contextual Foundations.....	208
Intuitive and Conceptual Underpinnings, Math Formulations	209
Masking Mechanisms	210
Causal Masking	211
Padding Masking	213
Custom Masks for Structured Data	214
Sparse Attention Masks.....	216
Learned and Adaptive Masks	218
Bidirectional and Cross-Attention Masks	219
Specialized Attention Forms.....	220
Challenges and Limitations of Masking Mechanisms	223
Complexity of Designing Masks	224
The Risk of Over-Masking	224
Computational Overhead of Generating and Applying Masks.....	225
Future Directions in Masking.....	228
The Enduring Legacy of SDPA.....	229
Custom Implementation of SDPA—Sliding Window and Grouped Query Attention for Our LLM.....	229
Understanding the Inputs and Outputs	232
Step 1: Shape Inference, Broadcasting, and Reshaping	233
Step 2: Computing Raw Attention Scores.....	234
Step 3: Constructing and Applying the Attention Mask	235
Step 4: Incorporating Sink Logits	236
Step 5: Softmax Normalization and Output Computation	237
Summary.....	238

Chapter 7: AttentionBlock with Rotary Embedding, GQA, Sliding Window, and Sink Tokens 241

- Fundamentals of Attention 241
- Self-Attention in Transformers 242
 - Operational Properties 243
- Causal Self-Attention 244
 - Implementation in Transformers 245
 - Role in LLMs 245
 - Limitations and Challenges 246
 - Optimizations and Variants 246
- Multihead Attention in Transformers 247
 - Operational Principles 248
 - Integration in Transformers 249
 - Positional Considerations 249
 - Role in LLMs 250
 - Limitations and Challenges 251
 - Optimizations and Variants 251
 - Advanced Considerations (up to 2025) 252
- What Is Grouped Query Attention, Sliding Window Attention, and Sink Tokens? 253
 - Grouped Query Attention (GQA) 253
 - Sliding Window Attention (SWA) 254
 - Sink Tokens 254
- Integration in the Attention Block 255
- Integration of Attention Mechanism in Our Custom Large Language Model 255
- Exhaustive Explanation of the AttentionBlock PyTorch Module 258
- Class Definition and PyTorch Integration 259
- Constructor (__init__) Method 259
 - Parameter Extraction and GQA Configuration 260
 - Validation Checks 260
 - Sliding Window and GQA Grouping 261
 - Normalization Layer 262

QKV Projection Layer	262
Output Projection Layer	263
Rotary Embedding Initialization	263
Sink Logits Parameter	264
Softmax Scaling Factor	265
Forward Pass (forward Method).....	265
Input Unpacking and Residual Connection	266
Pre-attention Normalization	266
QKV Projection and Slicing	266
GQA Reshaping	267
RoPE Application.....	267
Core Attention Computation.....	268
Output Projection and Residual Addition	269
Training and Inference Behaviors	270
Performance Optimizations.....	270
Interpretability.....	272
Comparison to Other Attention Mechanisms.....	273
Practical Implementation Notes.....	274
Summary.....	274
Chapter 8: Multilayer Perceptron Block with Mixture of Experts (MoE) and SwiGLU.....	277
Mixture of Experts: A Comprehensive Overview	278
Architecture of Mixture of Experts	279
Components	280
Sparse Activation.....	281
Variants of MoE	281
Mathematical Foundations	282
Training Mechanisms	283
Applications of Mixture of Experts.....	286
Advantages of MoE.....	286
Limitations of MoE.....	287

TABLE OF CONTENTS

SwiGLU: An In-Depth Exploration 288

 Historical Context 289

 The Basics: From Activation Functions to Gated Units 290

 Mathematical Representation 292

 Training Mechanisms 293

 Applications of SwiGLU..... 295

 Advantages of SwiGLU 295

 Limitations of SwiGLU 296

 Practical Considerations..... 296

 Advanced Variants and Comparisons 297

Multilayer Perceptron Blocks with Mixture of Experts (MoE) and SwiGLU: A Comprehensive Integration..... 298

 Integrated Architecture: MLP with MoE and SwiGLU..... 299

 Mathematical Foundations 301

 Training Mechanisms 302

 Why They Work Together: Rationale..... 304

 Advantages..... 305

 Limitations..... 305

 Practical Implementation Considerations..... 306

 Future Directions 307

 Multilayer Perceptron Block with Mixture of Experts (MoE) and SwiGLU for Our Large Language Model..... 308

In-Depth Analysis of the MLPBlock with Mixture of Experts (MoE) and SwiGLU..... 311

 Overview of the MLPBlock..... 311

 Forward Pass: Step-by-Step Execution 315

 Training Dynamics 321

 Edge Cases and Robustness 323

Summary..... 323

Chapter 9: Transformer Block and Full Transformer Model—It’s Time to Put the Puzzle Together	325
The Role of the Transformer Block in Sequence Processing	325
The Architecture of the Transformers Block.....	326
Query-Key-Value Projections and Multihead Decomposition	329
Rotary Position Embeddings: Geometric Foundations.....	330
Scaled Dot-Product Attention: The Core Mechanism.....	330
The Feed-Forward Block: Position-Wise Transformations.....	332
Alternative Activations: SwiGLU and Gated Variants.....	333
Block Composition and Information Flow.....	334
Full Model Architecture and Training Dynamics	335
Building the Transformers Block for Our LLM from Scratch	335
The TransformerBlock Class.....	337
Transformer Class—The Complete Language Model.....	341
from_checkpoint Class Method—Loading Pretrained Models.....	345
Architecture Design Choices and Modern Practices	350
Advantages, Challenges, and Broader Impact.....	351
Interpretability and Mechanistic Understanding.....	353
Fundamental Challenges	355
Future Trajectories.....	358
Summary	365
Chapter 10: Dataset Preparation, Model Training, Token Generator for Inference and Prompting—The BIG Moment.....	367
Dataset Preparation for LLM Training.....	367
The Importance of Dataset Quality	367
Data Collection and Sourcing	368
Data Cleaning and Filtering	369
Deduplication.....	370
Data Formatting and Tokenization.....	370
Dataset Composition and Mixing.....	371
Best Practices and Recommendations.....	372

TABLE OF CONTENTS

Preparing Our Dataset..... 374

 Text Characteristics and Classification of the Dataset 382

Data Preparation Pipeline for This Text 382

 Dataset Balancing Considerations..... 383

Training Large Language Models 384

 Model Architecture 385

 Training Objectives 385

 Optimization 385

 Distributed Training 385

 Infrastructure..... 385

 Training Stages..... 386

 Challenges 386

 Hyperparameter Selection..... 386

 Compute and Efficiency..... 386

 Advanced Techniques 387

 Post-training..... 387

 Ethical Considerations 387

Source Code for Training Our LLM 387

 Understanding the Code Step-by-Step..... 394

 Part 1: Environment Setup and Configuration 394

 Part 2: Dataset Preparation 396

 Part 3: Tokenizer Training or Loading..... 397

 Part 4: Dataset Encoding and Caching 398

 Part 5: DataLoader Setup..... 399

 Part 6: Model Configuration and Initialization..... 400

 Part 7: Optimizer Configuration 402

 Part 8: Learning Rate Scheduling 403

 Part 9: Model Compilation (Optional)..... 404

 Part 10: The Training Loop 404

 Part 11: Saving the Model 407

 Understanding Training Dynamics..... 408

 What the Model Learns..... 408

TokenGenerator for Inference	410
What Is TokenGenerator?	410
What Is TokenGenerator?	418
Core Functionality	418
The Generation Process	419
Sophisticated Anti-repetition System	419
Advanced Sampling Strategy	422
Why So Many Guardrails?	424
Practical Usage	424
Key Differences from Training	425
The Big Moment—Prompting Our Model	426
Understanding Inference Code and Model Behavior	430
Setup and Initialization	430
Building Anti-copy Indices	430
Text Cleaning Utilities	431
Loop Detection	431
Generation Function with Sophisticated Controls	432
Interactive Interface	432
Understanding the Output—An Educational Demonstration	433
The User Query	433
The Generated Output—Three Revealing Behaviors	433
What This Demonstrates About Language Model Design	436
Principle 1: Model Scale and Generalization	436
Principle 2: Training Data Diversity	437
Principle 3: Training Duration and Compute	438
Principle 4: Instruction Fine-Tuning and Alignment	439
The Value of This Demonstration	439
What Pretraining Alone Achieves	440
What Requires Additional Scale and Training	440
How Inference Mechanisms Work	440

TABLE OF CONTENTS

Practical Lessons from This Example 441

 Lesson 1: Match Model Scale to Task Complexity 441

 Lesson 2: Inference Constraints Have Limits..... 441

 Lesson 3: Training Data Determines Capabilities 441

 Lesson 4: Hyperparameter Tuning Matters..... 441

 Lesson 5: Multistage Training Is Essential 442

Exploring Even Further—A Real Trained Model: TinyStories GPT-4 Version Implementation.... 442

 TinyStories Dataset and Model..... 442

 Key Technical Differences from the Book’s Code 443

Why This Implementation Wasn’t Included in the Book 447

 The Reality of Training Costs 447

 Accessibility and Learning Goals..... 447

 The Scaling Gradient 448

 The Value Proposition 448

 When to Make the Investment..... 449

Appreciating What Was Achieved..... 449

 Looking Forward..... 450

Chapter 11: Advanced Training and CUDA Kernels 453

 The Journey from Raw Text to Intelligent Assistant—The Art and Science of LLM Training 453

 Pretraining—Building the Foundation 454

 Pretraining Data..... 455

 Next-Token Prediction 455

 Architecture and Training Setup 455

 Compute Budget, Duration, and Evaluation 456

 Mid-Training—Targeted Capability Development 456

 What Is Mid-Training?..... 456

 Why Mid-Training Matters 456

 Types of Mid-Training 457

 Mid-Training Methodology..... 458

 Preventing Catastrophic Forgetting..... 459

Examples of Successful Mid-Training	459
Mid-Training vs. Fine-Tuning	460
Supervised Fine-Tuning (SFT)—Teaching Instructions	460
The Transition from Base to Assistant	460
SFT Data: Instructions and Demonstrations	460
SFT Training Process	461
Data Quality in SFT	462
Balancing the SFT Dataset	462
Multiturn Dialog Training	463
The Result: An Instruction-Following Model	463
Reinforcement Learning from Human Feedback (RLHF)	464
The Alignment Problem	464
The Three Stages of RLHF	464
Reward Model Training	464
Reward Model Outputs	465
Reinforcement Learning Optimization	466
Challenges in RLHF	466
Practical Considerations	468
Results of RLHF	468
Alternative Alignment Approaches	469
Direct Preference Optimization (DPO)	469
Constitutional AI (CAI)	470
Reinforcement Learning from AI Feedback (RLAIF)	471
Iterative Approaches	471
Post-Training Techniques and Refinements	472
Context Distillation	472
Self-Improvement Techniques	473
Red Teaming and Adversarial Training	473
Capability-Specific Fine-Tuning	474
Multiobjective Optimization	474

TABLE OF CONTENTS

Evaluation and Benchmarking 474

 Evaluation During Pretraining..... 474

 Evaluation During Supervised Fine-Tuning..... 475

 Evaluation During Alignment Training..... 476

 Comprehensive Benchmarks..... 476

 The Limitations of Benchmarks..... 477

Practical Considerations and Best Practices 478

 Data Is King 478

 Computational Resource Management 478

 Preventing Degradation..... 479

 Scaling Laws and Efficiency..... 480

 Responsible AI Considerations 480

The Future of LLM Training 481

 Emerging Trends..... 481

 More Efficient Training Methods..... 482

 Better Evaluation 483

 Democratization 483

 Theoretical Understanding 484

Training Neural Networks with CUDA Kernels and Modern Frameworks..... 484

 Understanding CUDA and GPU Computing..... 485

 CUDA Kernels Explained 486

 CUDA Kernels in Neural Network Training 487

 Practical CUDA Kernel Examples 489

 Integration with Deep Learning Frameworks 498

 Performance Optimization Strategies..... 503

Chapter 7: Real-World Training Performance 505

 Practical Tips for Working with CUDA Kernels..... 507

Appendix: Glossary of Terms 510

Index..... 513