LLaMA-to-Titans Architecture Conversion Plan

Devin AI

January 16, 2025

1 Executive Summary

This document outlines the technical approach for converting the LLaMA 7B 3.3 transformer model to implement the Titans architecture. The conversion focuses on implementing Titans' three-component memory system while optimizing for specific hardware constraints (3x NVIDIA RTX 3090 GPUs, 64GB total VRAM).

2 Current Architecture (LLaMA 7B 3.3)

2.1 Core Components

- Model Parameters:
  - 32 transformer layers
  - 32 attention heads
  - Dimension size: 4096
  - Uses RMSNorm for layer normalization
  - Maximum sequence length: 4096 tokens
- Key Modules:
  - Attention mechanism with rotary positional embeddings (RoPE)
  - Parallel implementation via fairscale
  - Key-value caching for efficient inference
- Memory Management:
  - Current parameters: max_batch_size=32, max_seq_len=2048
  - Uses model parallel processing for memory distribution
  - Implements cache_k and cache_v for attention

3 Target Architecture (Titans)

3.1 Three-Component Memory System

- Core Module:
  - Modified attention mechanism from traditional transformers
  - Maintains accurate dependency modeling
  - Handles immediate context processing
- Long-term Memory:
  - Neural memory module for historical context
  - Specialized for maintaining long-range dependencies
  - Implements efficient retrieval mechanism
- Persistent Memory:
  - Task-specific knowledge storage
  - Optimized for specialized information retention
  - Supports 2M+ context window size

4 Implementation Strategy

4.1 Memory Distribution Across GPUs

Total VRAM: 64GB across 3x RTX 3090 GPUs

- GPU 1 (Core Module): ∼22GB
  - Primary attention mechanisms
  - Token embeddings
  - Layer normalization
- GPU 2 (Long-term Memory): ∼21GB
  - Historical context storage
  - Retrieval mechanisms
  - Cache management
- GPU 3 (Persistent Memory): ∼21GB
  - Task-specific knowledge base
  - Specialized storage systems
  - Integration logic

4.2 Architectural Modifications

- Core Module Adaptations:
  - Modify attention mechanism to support larger context windows
  - Implement efficient memory access patterns
  - Optimize for parallel processing
- Long-term Memory Implementation:
  - Design neural memory module
  - Implement retrieval mechanisms
  - Optimize for historical context maintenance
- Persistent Memory Integration:
  - Create specialized storage system
  - Implement knowledge integration logic
  - Optimize for task-specific retention

5 Testing and Validation

5.1 Unit Tests

- Core Module:
  - Attention mechanism functionality
  - Memory access patterns
  - Performance benchmarks
- Long-term Memory:
  - Historical context retention
  - Retrieval accuracy
  - Memory management efficiency
- Persistent Memory:
  - Knowledge storage integrity
  - Integration effectiveness
  - Task-specific performance

5.2 Integration Tests

- End-to-end system validation
- Performance metrics:
  - Context window size verification (2M+ tokens)
  - Memory usage monitoring
  - Processing speed benchmarks
- Cross-component interaction testing

6 Performance Optimization

6.1 Memory Management

- VRAM optimization strategies:
  - Gradient checkpointing
  - Memory-efficient attention patterns
  - Optimized cache management
- Load balancing across GPUs
- Memory access pattern optimization

6.2 Computational Efficiency

- Parallel processing optimization
- Cache utilization improvements
- Batch size optimization

7 Implementation Timeline

- Phase 1: Core Module Adaptation (1-2 weeks)
- Phase 2: Long-term Memory Implementation (1-2 weeks)
- Phase 3: Persistent Memory Integration (1-2 weeks)
- Phase 4: Testing and Optimization (1-2 weeks)

8 Conclusion

This implementation plan outlines the technical approach for converting LLaMA 7B 3.3 to the Titans architecture. The plan ensures efficient utilization of available hardware while maintaining model performance and implementing all key features of the Titans architecture.