MEGATRON

Large-Scale Language Model Training Framework

TRAINING FRAMEWORK [megatron/training/]

- Training Loop *training.py*
- Arguments *arguments.py*
- Checkpointing *checkpointing.py*
- Initialization *initialize.py*

MODELS

MCore

- GPT *core/models/gpt/*
- BERT *core/models/bert/*
- T5 *core/models/t5/*
- Mamba *core/models/mamba/*
- Multimodal *core/models/multimodal/*
- MoE *core/models/mimo/*

TRANSFORMER CORE

- **Attention** *core/transformer/attention.py* MCore
- **MLP** *core/transformer/mlp.py* MCore
- **Layer Norm** *core/transformer/* MCore
- **Embeddings** *core/transformer/* MCore
- **Block** *core/transformer/* MCore

PARALLELISM STRATEGY [megatron/core/]

TENSOR PARALLELISM

Intra-layer model sharding

core/tensor_parallel/ | layers.py | mappings.py

PIPELINE PARALLELISM

Inter-layer distribution

core/pipeline_parallel/ | schedules.py

DATA PARALLELISM

Gradient synchronization

core/distributed/ | ddp.py | fsdp.py

DATA PIPELINE

MCore

- Blended Dataset *core/datasets/blended_* MCore
- Indexed Dataset *core/datasets/indexed_* MCore
- GPT Dataset *core/datasets/gpt_* MCore
- Data Loader *training/datasets/*

OPTIMIZER

- Distributed Optimizer *core/optimizer/* MCore
- Adam Config *core/optimizer/* MCore
- Gradient Scaling *core/optimizer/* MCore

CHECKPOINTING [training/]

- Save/Load *training/*
- Distributed *core/dist_checkpointing/* MCore
- Model State *training/*

KEY FEATURES [megatron/core/]

PRECISION SUPPORT

MCore

• FP16 / BF16 Training
• FP8 (Hopper Optimized)
• FP4 Quantization
• Transformer Engine

MEMORY OPTIMIZATION

MCore

• Activation Checkpointing
• Sequence Parallelism
• Gradient Checkpointing
• Fine-grained Offloading

PERFORMANCE

MCore

• CUDA Graphs
• Fused Kernels
• Flash Attention
• Fused LayerNorm

Megatron-LM: Large-Scale Language Model Training Framework

NVIDIA | MCore Library | Distributed Training | Tensor | Pipeline | Data Parallelism

ARCHITECTURAL PATTERNS

Modular Core Library

MCore

megatron/core/ provides production-ready,
GPU-optimized building blocks for
framework developers

models | transformer | tensor_parallel
pipeline_parallel | distributed | optimizer

Configuration-Driven

Dataclass-based configuration system
enables flexible model parallelism
setup via command-line arguments

TransformerConfig | ModelParallelConfig
ProcessGroupCollection | parallel_state