

DeepSpeed Cheatsheet

Installation

pip install deepspeed
Install DeepSpeed library

pip install deepspeed-mii
Install DeepSpeed Model Implementations for Inference (MII)

Configuration

deepspeed --help
Display DeepSpeed CLI help

deepspeed --json_config=ds_config.json
Launch training with DeepSpeed configuration file

Training

deepspeed --num_gpus=4 train.py
Launch training script with 4 GPUs

deepspeed --num_nodes=2 --num_gpus=8 train.py
Distributed training across 2 nodes with 8 GPUs each

**model_engine, optimizer, _, _ =
deepspeed.initialize(args=args, model=model,
model_parameters=params)**
Initialize DeepSpeed engine in code

Monitoring

deepspeed --tensorboard_dir=./logs train.py
Enable TensorBoard logging

deepspeed --wandb train.py
Enable Weights & Biases integration

Checkpointing

model_engine.save_checkpoint(save_dir)
Save model checkpoint

**_, client_state =
model_engine.load_checkpoint(load_dir)**
Load model checkpoint

deepspeed --zero_stage=3 --save_interval=1
Save ZeRO-3 checkpoints every epoch

Inference

**model = deepspeed.init_inference(model,
mp_size=2, dtype=torch.half)**
Initialize model for inference with tensor parallelism

from deepspeed.mii import MIIserver
Import MII server for inference

MIIServer.start(model='gpt2', dtype='fp16')
Start inference server with a model

Profiling

deepspeed --autotuning=tune train.py
Enable autotuning for optimal performance

deepspeed --flops_profiler train.py
Enable FLOPS profiler

model_engine.flops_profiler.start_profile()
Start FLOPS profiling

model_engine.flops_profiler.stop_profile()
Stop FLOPS profiling and print results

ZeRO Optimization

deepspeed --zero_stage=1 train.py
Enable ZeRO stage 1 (optimizer state partitioning)

deepspeed --zero_stage=2 train.py
Enable ZeRO stage 2 (optimizer + gradient partitioning)

deepspeed --zero_stage=3 train.py
Enable ZeRO stage 3 (optimizer + gradient + parameter partitioning)

deepspeed --zero_offload train.py
Enable CPU offloading with ZeRO