
MLBench Benchmarks Documentation

MLBench development team

Nov 19, 2020

MLBENCH

1	MLBench Benchmark Implementations	1
1.1	Task 0: Communication Backend	1
1.2	Task 1: Image Classification	1
1.3	Task 2: Linear Learning	2
1.4	Task 3: Language Modelling	3
1.5	Task 4: Machine Translation	3
2	Indices and tables	5

MLBENCH BENCHMARK IMPLEMENTATIONS

MLBench contains several benchmark tasks and implementations. Tasks combinations of datasets and target metrics, whereas the implementations are concrete models and code that solve a task.

For an overview of MLBench tasks, please refer to the [Benchmarking Tasks Section](#)

A Benchmark Implementation is a model with fixed hyperparameters that solves a Benchmark Task.

1.1 Task 0: Communication Backend

This task is a dummy task that allows for testing the communication backends for various operations and frameworks.

1.1.1 0.a PyTorch All-reduce

PyTorch Communication backend Benchmarking

Benchmarking of communication backends in PyTorch

Task Task 0a

Framework PyTorch

Communication Backend Open MPI, GLOO, NCCL (PyTorch *torch.distributed*)

Distribution Algorithm All-Reduce

GPU Yes

Seed 42

Image Location mlbench/pytorch-backend-benchmark:latest

1.2 Task 1: Image Classification

1.2.1 1a. Resnet-20, CIFAR-10

PyTorch Cifar-10 ResNet-20 All-Reduce

Resnet 20 implementation for CIFAR-10 using All-Reduce

Task Task 1a

Framework PyTorch

Communication Backend Open MPI, GLOO, NCCL (PyTorch *torch.distributed*)

Distribution Algorithm All-Reduce

Model ResNet-20

Dataset CIFAR-10

GPU Yes

Seed 42

Image Location mlbench/pytorch-cifar10-resnet20-all-reduce:latest

PyTorch Cifar-10 ResNet-20 DDP

Resnet 20 implementation for CIFAR-10 using PyTorch DDP

Task Task 1a

Framework PyTorch

Communication Backend NCCL (PyTorch *torch.distributed*)

Distribution Algorithm Distributed Data Parallel

Model ResNet-20

Dataset CIFAR-10

GPU Yes

Seed 42

Image Location mlbench/pytorch-cifar10-resnet20-ddp:latest

1.2.2 1b. Resnet-?, ImageNet

TODO

1.3 Task 2: Linear Learning

1.3.1 2.a Logistic Regression, Epsilon 2008

PyTorch PASCAL Challenge 2008 Logistic Regression

Logistic Regression implementation

Task Task 2a

Framework PyTorch

Communication Backend Open MPI, GLOO (PyTorch *torch.distributed*)

Distribution Algorithm All-Reduce

Model Logistic Regression

Dataset PASCAL Challenge 2008 epsilon

GPU No

Seed 42

Image Location mlbench/pytorch-epsilon-logistic-regression-all-reduce:latest

1.4 Task 3: Language Modelling

1.4.1 3.a TODO

1.5 Task 4: Machine Translation

1.5.1 4.a LSTM, WMT16 EN-DE

PyTorch WMT16 GNMT Machine Translation

LSTM approach using GNMT implementation (adapted from [NVIDIA's implementation](#))

Task [Task 4a](#)

Framework PyTorch

Communication Backend Open MPI, NCCL and GLOO (PyTorch *torch.distributed*)

Distribution Algorithm All-Reduce

Model GNMT

Dataset WMT16

GPU Yes

Seed 42

Image Location mlbench/pytorch-wmt16-gnmt-all-reduce:latest

1.5.2 4.b Transformer, WMT17 EN-DE

PyTorch WMT17 Transformer Machine Translation

Transformer implementation [Attention Is All You need](#)

Task [Task 4b](#)

Framework PyTorch

Communication Backend Open MPI, NCCL and GLOO (PyTorch *torch.distributed*)

Distribution Algorithm All-Reduce

Model Transformer

Dataset WMT17

GPU Yes

Seed 42

Image Location mlbench/pytorch-wmt17-transformer-all-reduce:latest

INDICES AND TABLES

- genindex
- modindex
- search