



寒武纪 PyTorch 版本说明书

2021 年 10 月 19 日



目录

目录	i
表格目录	1
1 模块概述	2
2 v1.0.0	3
2.1 测试过的版本配置组合	3
2.2 Breaking Changes	4
2.3 特性变更	4
2.4 已修复问题	8
2.5 已知遗留问题	8
3 v0.19.2	9
3.1 测试过的版本配置组合	9
3.2 特性变更	9
3.3 已修复问题	13
3.4 已知遗留问题	14
4 v0.19.1	15
4.1 测试过的版本配置组合	15
4.2 Breaking Changes	15
4.3 特性变更	16
4.4 已修复问题	19
4.5 已知遗留问题	19



表格目录

2.1	测试过的版本组合 v1.0.0	3
2.2	v1.0.0 网络精度性能信息	4
3.1	测试过的版本组合 v0.19.2	9
3.2	v0.19.2 网络精度性能信息	10
4.1	测试过的版本组合 v0.19.1	15
4.2	v0.19.1 网络精度性能信息	16



1 模块概述

PyTorch 是 Facebook 开源的深度学习编程框架，适用于 Python、C++ 等编程语言，用以实现高效的 GPU 并行计算及深度学习网络搭建，具有轻松扩展、快速实现、生产部署稳定性强等特点。

为支持寒武纪 MLU（Machine Learning Unit，机器学习处理器），寒武纪定制了 PyTorch（以下简称 Cambricon PyTorch）。

Cambricon PyTorch 兼容原生 PyTorch 的 Python 编程接口和原生 PyTorch 网络模型，支持单机单卡训练和单机多卡训练功能。应用程序开发人员可以使用 Python 接口来完成各种正向推理任务。正向推理时，网络权重可以从 pth 格式文件中读取，已支持的分类和检测网络结构由 torchvision 管理，可以从 torchvision 中读取。

应用程序开发人员可以使用 Python 接口完成各种单机单卡训练任务，也可以使用 PyTorch 原生分布式机制 Distributed Data Parallel（DDP）进行单机多卡训练，并支持寒武纪自研通信后端 CNCL。

2.1 测试过的版本配置组合

表 2.1: 测试过的版本组合 v1.0.0

Cambricon PyTorch	1.0.0
PyTorch	1.6
Python	3.6, 3.7
Driver	4.15.9
CNToolkit	2.5.2
CNNL	1.7.1
CNCL	0.11.1
CNMIX	0.6.1
CNLight	0.8.0
Tested distributions	Ubuntu 16.04 (GCC 5.4.0)

2.2 Breaking Changes

- 新增支持 MMCV、MMDetection、MMAAction 模型仓库。
- 基于 MMDetection 框架，新增网络支持：TSN、RetinaNet、Faster R-CNN、Mask R-CNN。

2.3 特性变更

注意:

本版本专门用于使用 CNNL 进行训练的场景。

- 新增《寒武纪 PyTorch 性能调优手册》。
- 新增网络支持 Tacotron2、WaveGlow、CRF。
- 新增支持 index_copy_、inverse、logdet 算子。详细说明，参见《寒武纪 PyTorch 用户手册》。
- 支持 MMCV 框架自有算子 sigmoid_focal_loss、RoIAlign、nms。
- 提升 index、index_put、bmm、mm、linear、matmul 算子性能，优化 Device2Host 拷贝性能。
- Notify 缓存优化，提升训练网络性能。
- 训练支持 torch.manual_seed，torch.manual_seed_all。
- 完善 interpolate、equal、AvgPool3d、binary_cross_entropy、binary_cross_entropy_with_logits、index_copy_、index_select、svd、addmm_、repeat、eye、flip、conv1d、unflod 算子支持，更新算子规格限制。详细说明，参见《寒武纪 PyTorch 用户手册》。
- 更新部分网络的训练精度和性能信息，参见 v1.0.0 网络精度性能信息。

Cambricon PyTorch 训练后会得到浮点模型和相关的定点量化参数。下表中的定点、浮点表示的是 Cambricon PyTorch 训练后的两种推理模式。定点推理指利用量化参数对浮点模型进行量化并在 MLU 上直接进行定点量化推理。浮点推理指忽略量化参数，直接将浮点模型转换为 CPU 的浮点模型格式，然后在 CPU 上进行浮点推理。具体方法，参见《寒武纪 PyTorch 用户手册》的“模型训练”章节。

表 2.2: v1.0.0 网络精度性能信息

网络名称	数据集	精度	模式	单卡性能	4 卡性能	4 卡- 扩展性	8 卡性能	8 卡- 扩展性
AlexNet	ImageNet2012	N/A	FP32	bs: 512 2250.9fps	bs: 512 3520.8fps	39.1%	bs: 512 3666.9fps	20.36%
BBN	ucf101	44.8	FP32	bs: 208 160.0fps	bs: 208 153.4fps	23.97%	bs: 208 151.3fps	11.82%

下页继续

表 2.2 – 续上页

网络名称	数据集	精度	模式	单卡性能	4 卡性能	4 卡- 扩 展 性	8 卡性能	8 卡- 扩 展 性
DenseNet201	ImageNet2012	78.84- @140epoch	FP32	bs: 96 150.1fps	bs: 96 588.6fps	98.03%	bs: 96 1154.0fps	96.1%
GoogleNet	ImageNet2012	68.896	FP32	bs: 256 660.7fps	bs: 256 2422.8fps	91.68%	bs: 256 3551.1fps	67.18%
InceptionV2	ImageNet2012	N/A	FP32	bs: 512 586.8fps	bs: 512 2175.0fps	92.66%	bs: 512 3213.1fps	68.45%
InceptionV3	ImageNet2012	N/A	FP32	bs: 216 219.9fps	bs: 216 866.9fps	98.56%	bs: 216 1719.2fps	97.73%
MobileNetV2	ImageNet2012	71.894	FP32	bs: 256 647.5fps	bs: 256 2505.4fps	178.96%	bs: 256 3555.7fps	126.99%
MobileNetV2	ImageNet2012	N/A	Mixed	bs: 256 350.0fps	bs: 256 2505.4fps	178.96%	bs: 256 3555.7fps	126.99%
NGC_ResNet50- v1.5	ImageNet2012	N/A	FP32	bs: 256 350.5fps	bs: 256 1328.6fps	94.76%	bs: 256 2643.2fps	94.27%
P3D	ImageNet2012	0.7984	FP32	bs: 48 54.3fps	bs: 48 204.3fps	94.06%	bs: 48 405.6fps	93.37%
ResNet101	ImageNet2012	N/A	FP32	bs: 168 197.8fps	bs: 168 770.0fps	97.32%	bs: 168 1528.2fps	96.57%
ResNet18	ImageNet2012	N/A	FP32	bs: 664 885.5fps	bs: 664 2165.9fps	61.15%	bs: 664 3458.9fps	48.83%
ResNet50_v1.5	ImageNet2012	N/A	FP32	bs: 256 339.1fps	bs: 256 2372.8fps	171.49%	bs: 256 3605.0fps	130.28%
ResNet50_v1.5	ImageNet2012	N/A	Mixed	bs: 256 345.9fps	bs: 256 2372.8fps	171.49%	bs: 256 3605.0fps	130.28%

下页继续

表 2.2 – 续上页

网络名称	数据集	精度	模式	单卡性能	4 卡性能	4 卡- 扩展性	8 卡性能	8 卡- 扩展性
Shufflenet-v2_x0_5	ImageNet2012	N/A	FP32	bs: 1024 2041.0fps	bs: 1024 2548.5fps	31.22%	bs: 1024 3752.3fps	22.98%
Shufflenet-v2_x1_0	ImageNet2012	N/A	FP32	bs: 976 1220.4fps	bs: 976 3579.2fps	73.32%	bs: 976 4041.8fps	41.4%
Shufflenet-v2_x1_5	ImageNet2012	N/A	FP32	bs: 512 938.0fps	bs: 512 3339.8fps	89.01%	bs: 512 3641.8fps	48.53%
VGG16	ImageNet2012	N/A	FP32	bs: 128 148.3fps	bs: 128 581.2fps	97.98%	bs: 128 1120.0fps	94.4%
VGG16_bn	ImageNet2012	N/A	FP32	bs: 128 129.4fps	bs: 128 507.7fps	98.09%	bs: 128 980.5fps	94.72%
VGG19	ImageNet2012	N/A	FP32	bs: 64 115.0fps	bs: 64 452.2fps	98.3%	bs: 64 847.8fps	92.15%
Bert_base_fine-tune_msra_ner	msra	N/A	FP32	bs: 40 91.1 sequences/sec	bs: 40 362.0 sequences/sec	99.34%	bs: 40 702.3 sequences/sec	96.36%
ModelZoo-BERT	SQUAD1.1	88.378	FP32	bs: 32 39.3 sequences/sec	bs: 32 216.4 sequences/sec	99.63%	bs: 32 420.6 sequences/sec	96.82%
ModelZoo-BERT	SQUAD1.1	N/A	Mixed	bs: 32 54.3 sequences/sec	bs: 32 216.4 sequences/sec	99.63%	bs: 32 420.6 sequences/sec	96.82%
NGC-BERT	SQUAD1.1	N/A	FP32	bs: 24 28.4 sequences/sec	bs: 24 113.3 sequences/sec	99.74%	bs: 24 227.0 sequences/sec	99.91%
Transformer	IWSLT2016	16.335- @20epoch	FP32	bs: 1984 22542.0fps	bs: 1984 67311.2fps	74.65%	bs: 1984 87205.0fps	48.36%

下页继续

表 2.2 – 续上页

网络名称	数据集	精度	模式	单卡性能	4 卡性能	4 卡- 扩展性	8 卡性能	8 卡- 扩展性
Transformer-fairseq	wmt17	27.91- @100epoch	FP32 (BLEU4)	bs: -1 7759.2wps	bs: -1 45384.4wps	95.31%	bs: -1 74218.2wps	77.93%
Transformer-fairseq	wmt17	N/A	Mixed	bs: -1 11903.9wps	bs: -1 45384.4wps	95.31%	bs: -1 74218.2wps	77.93%
CenterNet	COCO2017	N/A	FP32	bs: 40 3.4fps	bs: 40 13.5fps	99.26%	bs: 40 27.1fps	99.63%
FasterRCNN-ResNet101+FPN	COCO2017	N/A	FP32	bs: 8 2.3fps	bs: 8 8.8fps	95.65%	bs: 8 16.4fps	89.13%
MaskRCNN-ResNet101+FPN	COCO2017	39.89(bbox) 36.08(segm)	FP32	bs: 4 1.6fps	bs: 4 5.7fps	89.06%	bs: 4 9.6fps	75.0%
RFBNet	VOC2007	N/A	FP32	bs: 64 35.1fps	bs: 64 134.8fps	96.01%	bs: 64 252.8fps	90.03%
SSD-ResNet50	COCO2017	N/A	FP32	bs: 80 71.0fps	bs: 80 277.4fps	97.68%	bs: 80 558.4fps	98.31%
SSD-VGG16	VOC	N/A	FP32	bs: 18 27.7fps	bs: 18 109.1fps	98.47%	bs: 18 202.1fps	91.2%
YOLOv3	COCO2014	N/A	FP32	bs: 16 37.0fps	bs: 16 139.1fps	93.99%	bs: 16 261.4fps	88.31%
YOLOv5-m	COCO2017	0.379	FP32	bs: 88 130.9fps	bs: 88 458.2fps	87.51%	bs: 88 575.1fps	54.92%
YOLOv5-s	COCO2017	0.505	FP32	bs: 80 57.1fps	bs: 80 192.5fps	84.28%	bs: 80 348.1fps	76.2%
CRNN	Synth90k	0.885223	FP32	bs: 128 2670.5fps	bs: 128 15288.3fps	143.12%	bs: 128 39234.6fps	183.65%

2.4 已修复问题

- 修复 ger 算子计算错误的问题。
- 解除 index_put 算子不支持 accumulate 参数为 True 的限制。
- 修复 Ubuntu 1804 docker 上 torch.nn.Upsample(mode='bilinear') 算子报错的问题。

2.5 已知遗留问题

- 框架中对 int64 和 Double 数据类型进行了截断处理。

现象：

PyTorch 中 Long 或 int64 类型数据被截断为 int32，Double 类型数据被截断为 float 类型。

影响：

对一些使用 int64 以及 Double 类型数据的网络可能会造成精度影响。

规避措施：

无规避措施。

- 训练时，如果 optimizer 的构造放在 model 拷贝到 MLU 之前，loss 不下降。

现象：

在训练网络时，如果将 optimizer 的构造放在 model 拷贝到 MLU 之前，loss 没有下降。

影响：

网络无法正常收敛，精度不达标。

规避措施：

把 model 的拷贝放在 optimizer 构造之前。

3.1 测试过的版本配置组合

表 3.1: 测试过的版本组合 v0.19.2

Cambricon PyTorch	0.19.2
PyTorch	1.6
Python	3.6, 3.7
Driver	4.15.7
CNToolkit	2.3.4
CNNL	1.5.4
CNCL	0.10.3
CNMIX	0.5.2
Tested distributions	Ubuntu 16.04 (GCC 5.4.0)

3.2 特性变更

注意:

本版本专门用于使用 CNNL 进行训练的场景。因本版本无法保证使用 CNML 进行网络推理的精度和性能，因此不建议使用该版本调用 CNML 进行网络推理。

- 分类网络支持 dummy test，用于性能测试。
- 停止对 CNML 算子库后端的支持。

- 完善 Pytorch 一键编译脚本。
- 发布《寒武纪 PyTorch 性能测试手册》。
- 废弃 adaptive_quantize 自适应量化接口，通过 CNMIX 实现混合精度训练，详细参考：《寒武纪 PyTorch 网络移植手册》。
- 更新部分网络的训练精度和性能信息，参见 v0.19.2 网络精度性能信息。

Cambricon PyTorch 训练后会得到浮点模型和相关的定点量化参数。下表中的定点、浮点表示的是 Cambricon PyTorch 训练后的两种推理模式。定点推理指利用量化参数对浮点模型进行量化并在 MLU 上直接进行定点量化推理。浮点推理指忽略量化参数，直接将浮点模型转换为 CPU 的浮点模型格式，然后在 CPU 上进行浮点推理。具体方法，参见《寒武纪 PyTorch 用户手册》的“模型训练”章节。

表 3.2: v0.19.2 网络精度性能信息

网络名称	数据集	精度	模式	性能	4 卡性能	4 卡- 扩展性	8 卡性能	8 卡- 扩展性
ResNet50_v1.5	ImageNet2012	N/A	FP32	bs: 256 322.5fps	bs: 256 1237.8fps	95.9%	bs: 256 2409.0fps	93.4%
ResNet50_v1.5	ImageNet2012	N/A	Mixed	bs: N/A N/A	bs: N/A N/A	N/A	bs: N/A N/A	N/A
ResNet18	ImageNet2012	N/A	FP32	bs: 664 845.5fps	bs: 664 N/A	N/A	bs: 664 N/A	N/A
NGC_ResNet50-v1.5	ImageNet2012	N/A	FP32	bs: 256 326.0fps	bs: 256 N/A	N/A	bs: 256 N/A	N/A
NGC_ResNet50-v1.5	ImageNet2012	N/A	Mixed	bs: N/A N/A	bs: N/A N/A	N/A	bs: N/A N/A	N/A
AlexNet	ImageNet2012	N/A	FP32	bs: 256 1394.9fps	bs: 2768 N/A	N/A	bs: 2768 N/A	N/A
VGG16	VOC2007	N/A	FP32	bs: 160 141.5fps	bs: 160 N/A	N/A	bs: 160 N/A	N/A
VGG16_bn	ImageNet2012	N/A	FP32	bs: 128 120.4fps	bs: 128 N/A	N/A	bs: 128 N/A	N/A

下页继续

表 3.2 – 续上页

网络名称	数据集	精度	模式	性能	4 卡性能	4 卡- 扩展性	8 卡性能	8 卡- 扩展性
InceptionV3	ImageNet2012	N/A	FP32	bs: 216 190.0fps	bs: 216 N/A	N/A	bs: 216 N/A	N/A
MobileNetV2	ImageNet2012	71.894	FP32	bs: 416 573.0fps	bs: 416 2123.9wps	92.6%	bs: 416 4052.4	88.4%
MobileNetV2	ImageNet2012	71.684	Mixed	bs: 416 572.4fps	bs: 416 N/A	N/A	bs: 416 N/A	N/A
GoogleNet	ImageNet2012	N/A	FP32	bs: 520 586.0fps	bs: 520 N/A	N/A	bs: 520 N/A	N/A
ResNet101	ImageNet2012	N/A	FP32	bs: 168 190.3fps	bs: 168 N/A	N/A	bs: 168 N/A	N/A
VGG19	ImageNet2012	N/A	FP32	bs: 160 114.3fps	bs: 160 N/A	N/A	bs: 160 N/A	N/A
InceptionV2	ImageNet2012	N/A	FP32	bs: 512 506.0fps	bs: 512 N/A	N/A	bs: 512 N/A	N/A
Shufflenet-v2_x0_5	ImageNet2012	N/A	FP32	bs: 1872 1219.8fps	bs: 1872 N/A	N/A	bs: 1872 N/A	N/A
Shufflenet-v2_x1_0	ImageNet2012	N/A	FP32	bs: 976 1092.5fps	bs: 976 N/A	N/A	bs: 976 N/A	N/A
Shufflenet-v2_x1_5	ImageNet2012	N/A	FP32	bs: 720 886.3fps	bs: 784 N/A	N/A	bs: 784 N/A	N/A
DenseNet201	ImageNet2012	N/A	FP32	bs: 96 63.8fps	bs: 96 N/A	N/A	bs: 96 N/A	N/A
Transformer-fairseq	wmt17	27.91- @100epoch BLEU4	FP32	bs: -1 7267.9wps	bs: -1 28104.5wps	96.6%	bs: -1 52302.0	89.9%

下页继续

表 3.2 – 续上页

网络名称	数据集	精度	模式	性能	4 卡性能	4 卡- 扩 展 性	8 卡性能	8 卡- 扩 展 性
Transformer-fairseq	wmt17	N/A	Mixed	bs: -1 N/A	bs: -1 N/A	N/A	bs: -1 N/A	N/A
Transformer	IWSLT2016	N/A	FP32	bs: 2048 15656.9fps	bs: 2048 N/A	N/A	bs: 2048 N/A	N/A
ModelZoo-BERT	SQUAD1.1	88.378	FP32	bs: 32 36.1	bs: 32 144.0	99.7%	bs: 32 282.0	97.6%
ModelZoo-BERT	SQUAD1.1	88.199	Mixed	bs: N/A N/A	bs: N/A N/A	N/A	bs: N/A N/A	N/A
NGC-BERT	SQUAD1.1	N/A	FP32	bs: 24 24.8 seq/s	bs: 24 N/A	N/A	bs: 24 N/A	N/A
Bert_base_fine-tune_msra_ner	msra	N/A	FP32	bs: 40 64.2 seq/s	bs: 40 N/A	N/A	bs: 40 N/A	N/A
SSD-ResNet50	COCO2017	N/A	FP32	bs: 80 73.1fps	bs: 80 N/A	N/A	bs: 80 N/A	N/A
SSD-VGG16	VOC	N/A	FP32	bs: 32 26.6fps	bs: 32 N/A	N/A	bs: 32 N/A	N/A
YOLOv3	COCO2014	N/A	FP32	bs: 16 27.4fps	bs: 16 N/A	N/A	bs: 16 N/A	N/A
YOLOv5-s	COCO2017	0.505	FP32	bs: 16 34.8fps	bs: 16 N/A	N/A	bs: 16 N/A	N/A
YOLOv5-m	COCO2017	N/A	FP32	bs: 176 142.1fps	bs: 176 N/A	N/A	bs: 176 N/A	N/A

下页继续

表 3.2 – 续上页

网络名称	数据集	精度	模式	性能	4 卡性能	4 卡- 扩展性	8 卡性能	8 卡- 扩展性
RetinaNet	COCO2017	N/A	FP32	bs: 8 1.898fps	bs: 8 N/A	N/A	bs: 8 N/A	N/A
MaskRCNN- ResNet101+FPN	COCO2017	39.89(bbox) 36.08(seg)	FP32	bs: 2 1.1fps	bs: 2 N/A	N/A	bs: 2 N/A	N/A
FasterRCNN- ResNet101+FPN	COCO2017	N/A	FP32	bs: 8 1.7fps	bs: 14 N/A	N/A	bs: 8 N/A	N/A
BBN	UCF101	44.8	FP32	bs: 208 307.0fps	bs: 208 N/A	N/A	bs: 208 N/A	N/A
P3D	ImageNet2012	N/A	FP32	bs: 16 46.1fps	bs: 48 N/A	N/A	bs: 48 N/A	N/A
RFBNet	VOC2007	0.7984	FP32	bs: 88 31.6fps	bs: 88 N/A	N/A	bs: 88 N/A	N/A
CenterNet	COCO2017	N/A	FP32	bs: 40 3.3fps	bs: 40 N/A	N/A	bs: 40 N/A	N/A
CRNN	Synth90k	N/A	FP32	bs: 128 2726.5	N/A	bs: 128 N/A	bs: 128 N/A	N/A

3.3 已修复问题

- 修复 nllloss 算子不连续输入计算出错的问题。
- 修复 index_select 不支持 bool 类型输入的问题。
- 修复 dcn 算子计算错误的问题。
- 修复 embedding 算子不支持一维输入的问题。
- 修复逐层 dump 工具不连续输入导致出错的问题。
- 修复 lstm 小算子拼接实现计算错误的问题。

3.4 已知遗留问题

- 框架中对 int64 和 Double 数据类型进行了截断处理。

现象：

PyTorch 中 Long 或 int64 类型数据被截断为 int32，Double 类型数据被截断为 float 类型。

影响：

对一些使用 int64 以及 Double 类型数据的网络可能会造成精度影响。

规避措施：

无规避措施。

- 在 Ubuntu 1804 docker 上运行 upsample_bilinear2d 算子会报错。

现象：

在 Ubuntu 1804 docker 上运行算子 `torch.nn.Upsample(mode='bilinear')` 时，会报错 `Segmentation fault(core dumped)`。

原因：

Ubuntu 1804 docker 中的底层工具库问题导致。

影响：

Ubuntu 1804 docker 中无法运行算子 `torch.nn.Upsample(mode='bilinear')`。

规避措施：

使用 Ubuntu 1604 docker。

- `index_put` 算子不支持 `accumulate` 参数为 `True` 的场景。

现象：

如果 `index_put` 算子的 `accumulate` 参数设置为 `True`，该算子会转移到 CPU 上运行。

原因：

CNNL 算子库暂不支持该场景。

影响：

`index_put` 算子 `accumulate` 参数为 `True` 的场景会运行到 CPU 上，影响网络性能。

规避措施：

无。

4.1 测试过的版本配置组合

表 4.1: 测试过的版本组合 v0.19.1

Cambricon PyTorch	0.19.1
PyTorch	1.6
Python	3.6, 3.7
Driver	4.15.4
CNToolkit	2.3.3
CNNL	1.5.3
CNCL	0.10.2
CNMIX	0.5.1
Tested distributions	Ubuntu 16.04 (GCC 5.4.0)

4.2 Breaking Changes

- 支持 TENSOR VIEWS 功能，支持 tensor 视图功能，共享物理内存。
- 支持 Memory Format 功能，支持 tensor Channels Last 功能。

4.3 特性变更

注意:

本版本专门用于使用 CNML 进行训练的场景。因本版本无法保证使用 CNML 进行网络推理的精度和性能，因此不建议使用该版本调用 CNML 进行网络推理。

- 新增网络支持：P3D、BBN、RFBNet、CenterNet、YOLOv5-m、CRNN。
- 新增算子支持：mseloss、baddbmm、lstm、narrow、broadcast_tensors、dcn、l1_loss。详细说明，参见《寒武纪 PyTorch 用户手册》。
- 进一步优化 NLP 相关网络性能。详细说明，参见 v0.19.1 网络精度性能信息。
- MLU Profiler 工具新增 `profile_memory` 参数可用于内存使用情况分析。
- 更新部分网络的训练精度和性能信息。

Cambricon PyTorch 训练后会得到浮点模型和相关的定点量化参数。下表中的定点、浮点表示的是 Cambricon PyTorch 训练后的两种推理模式。定点推理指利用量化参数对浮点模型进行量化并在 MLU 上直接进行定点量化推理。浮点推理指忽略量化参数，直接将浮点模型转换为 CPU 的浮点模型格式，然后在 CPU 上进行浮点推理。具体方法，参见《寒武纪 PyTorch 用户手册》的“模型训练”章节。

表 4.2: v0.19.1 网络精度性能信息

网络名称	数据集	精度	模式	性能 (MLU370-X8 单芯片)	性能 (MLU370-X8 整卡)
ResNet50_v1.5	ImageNet2012	76.440@100epoch	FP32	batch_size: 256 163.1fps	batch_size: 256 322.1fps
ResNet18	ImageNet2012	N/A	FP32	batch_size: 512 433.7fps	batch_size: 512 846.8fps
NGC_ResNet50-v1.5	ImageNet2012	N/A	FP32	batch_size: 256 164.7fps	batch_size: 256 325.4fps
AlexNet	ImageNet2012	N/A	FP32	batch_size: 256 705.3fps	batch_size: 256 1394.9fps
VGG16	ImageNet2012	N/A	FP32	batch_size: 128 71.4fps	batch_size: 128 141.4fps

下页继续

表 4.2 – 续上页

网络名称	数据集	精度	模式	性能 (MLU370-X8 单芯片)	性能 (MLU370-X8 整卡)
VGG16_bn	ImageNet2012	N/A	FP32	batch_size: 128 61.5fps	batch_size: 128 121.5fps
InceptionV3	ImageNet2012	N/A	FP32	batch_size: 128 97.8fps	batch_size: 128 195.7fps
MobileNetV2	ImageNet2012	N/A	FP32	batch_size: 256 88.2fps	batch_size: 256 174.7fps
GoogleNet	ImageNet2012	N/A	FP32	batch_size: 128 285.7fps	batch_size: 128 556.5fps
ResNet101	ImageNet2012	75.362	FP32	batch_size: 128 97.1fps	batch_size: 128 191.3fps
VGG19	ImageNet2012	N/A	FP32	batch_size: 128 57.8fps	batch_size: 128 114.8fps
InceptionV2	ImageNet2012	N/A	FP32	batch_size: 256 261.2fps	batch_size: 256 513.1fps
Shufflenet-v2_x0_5	ImageNet2012	N/A	FP32	batch_size: 128 534.2fps	batch_size: 128 1025.8fps
Shufflenet-v2_x1_0	ImageNet2012	N/A	FP32	batch_size: 128 282.9fps	batch_size: 128 553.5fps
Shufflenet-v2_x1_5	ImageNet2012	N/A	FP32	batch_size: 128 208.4fps	batch_size: 128 409.6fps
DenseNet201	ImageNet2012	N/A	FP32	batch_size: 32 29.8fps	batch_size: 32 58.7fps
Transformer-fairseq	训练: WMT16ENDE 测试: new-stest2014	27.91(BLEU4)@100epoch	FP32	3699.4 words/s	7028.1 words/s

下页继续

表 4.2 – 续上页

网络名称	数据集	精度	模式	性能 (MLU370-X8 单芯片)	性能 (MLU370-X8 整卡)
Transformer	IWSLT16	N/A	FP32	batch_size: 32 1041.8 words/s	batch_size: 32 2859.3 words/s
ModelZoo-BERT	Squad V1.1	88.378	FP32	batch_size: 16 17.4 sequence/sec	batch_size: 16 34.2 sequence/sec
ModelZoo-BERT	Squad V1.1	88.199	Mixed	N/A	N/A
NGC-BERT	Squad V1.1	N/A	FP32	batch_size: 16 12.7 sequence/sec	batch_size: 16 24.9 sequence/sec
Bert_base_fine-tune_msra_ner	MSRA	N/A	FP32	batch_size: 30 45.8 sequence/sec	batch_size: 30 64.2 sequence/sec
SSD-ResNet50	COCO2017	N/A	FP32	batch_size: 32 36.2fps	batch_size: 32 71.7fps
SSD-VGG16	VOC2007	N/A	FP32	batch_size: 32 13.8fps	batch_size: 32 26.6fps
YOLOv3	COCO2014	N/A	FP32	batch_size: 16 13.9fps	batch_size: 16 27.4fps
YOLOv5-s	COCO2017	0.505	FP32	batch_size: 16 17.6fps	batch_size: 16 34.8fps
YOLOv5-m	COCO2017	N/A	FP32	batch_size: 40 46.1fps	batch_size: 40 89.6fps
RetinaNet	COCO2017	N/A	FP32	batch_size: 4 1.75fps	batch_size: 4 2.28fps

下页继续

表 4.2 – 续上页

网络名称	数据集	精度	模式	性能 (MLU370-X8 单芯片)	性能 (MLU370-X8 整卡)
MaskRCNN-ResNet101+FPN	COCO2017	N/A	FP32	batch_size: 2 0.6fps	batch_size: 2 1.1fps
FasterRCNN-ResNet101+FPN	COCO2017	N/A	FP32	batch_size: 8 0.9fps	batch_size: 8 1.7fps
BBN	UCF101	44.8	FP32	batch_size: 128 18.4fps	batch_size: 128 36.8fps
P3D	UCF101	N/A	FP32	batch_size: 16 23.6fps	batch_size: 16 46.1fps
RFBNet	VOC2007	N/A	FP32	batch_size: 32 15.2fps	batch_size: 32 28.4fps
CenterNet	COCO2017	N/A	FP32	batch_size: 16 1.6fps	batch_size: 16 3.2fps
CRNN	Synth90K	N/A	FP32	batch_size: 64 581.8fps	batch_size: 64 1142.9fps

4.4 已修复问题

- 修复 MLU Profiler 工具由于已知的底层库问题，在特定情况下无法采集 MLU 的性能数据问题。
- 修复不支持 shared_storage 功能导致的占用额外的设备内存问题。
- 修复原位计算时，生成一个新 tensor，并且该 tensor 会替换原有 tensor 的 data_ptr 和 data_type，以及 offset 和 stride 未被正确设置问题。

4.5 已知遗留问题

- 框架中对 int64 和 Double 数据类型进行了截断处理。

现象：

PyTorch 中 Long 或 int64 类型数据被截断为 int32，Double 类型数据被截断为 float 类型。

影响：

对一些使用 int64 以及 Double 类型数据的网络可能会造成精度影响。

规避措施：

无规避措施。

- 在 pytorch_models 仓库运行多卡网络时卡死。

现象：

运行 pytorch_models 仓库中的多卡网络会出现卡死现象。

原因：

未设置环境变量 MLU_VISIBLE_DEVICES。

影响：

无法使用多卡运行网络脚本。

规避措施：

设置环境变量 MLU_VISIBLE_DEVICES。例如，要使用 0、1、2、3 卡运行 4 卡网络，需要执行 `export MLU_VISIBLE_DEVICES=0,1,2,3`。

- 安装了 catch wheel 包后运行报错。

现象：

安装了 catch wheel 包后运行时报错 `ImportError: No module named 'xxx'` 或者 `ModuleNotFoundError: No module named 'xxx'`。

原因：

没有安装 catch 的依赖包。

影响：

只执行 catch wheel 包安装无法运行 Cambricon Pytorch。

规避措施：

使用源码编译安装或者安装完 catch 所有依赖 wheel 包，或者直接使用 docker 运行。

- 在 Ubuntu 1804 docker 上运行 upsample_bilinear2d 算子会报错。

现象：

在 Ubuntu 1804 docker 上运行算子 `torch.nn.Upsample(mode='bilinear')` 时，会报错 `Segmentation fault(core dumped)`。

原因：

Ubuntu 1804 docker 中的底层工具库问题导致。

影响：

Ubuntu 1804 docker 中无法运行算子 `torch.nn.Upsample(mode='bilinear')`。

规避措施：

使用 Ubuntu 1604 docker。

- index_put 算子不支持 accumulate 参数为 True 的场景。

现象：

如果 index_put 算子的 accumulate 参数设置为 True，该算子会转移到 CPU 上运行。

原因：

CNNL 算子库暂不支持该场景。

影响：

index_put 算子 accumulate 参数为 True 的场景会运行到 CPU 上，影响网络性能。

规避措施：

无。