mmlab_extension

发行版本 *latest*

windzu

2022 年 09 月 03 日

Get Started

CHAPTER 1

引言

作者本人目前为自动驾驶感知方向，所以功能实现时候会优先自动驾驶方向

一个深度学习任务的闭环简单可以分为如下三个部分

- 数据：采集、标注、数据管理
- 模型：网络搭建、训练、测试、评估
- 部署：模型部署

随着任务复杂性的提高，依赖单人完成全流程工作变的困难起来，而如果依赖多人的合作，又往往会出现因为水平不匹配、工作内容冲突带来的合作者间的“扯皮”问题，针对此情况，业界提出了 `MLOps` 的概念

但目前就本人调研来看，还没有成熟的 `MLOps` 解决方案，众多开源工具也是限制颇多，所以在没有成熟的好用工具之前，只能自己利用开源工具简单你的拼凑一个出来使用了 ~

本工程是依赖open-mmlab中的一系列工具完成了 MLOps 任务中的 modeling 和 deployment 的工作，因为其中所主要使用的 mmdeploy 和 mmdetection3d 还在快速的更新，所以想使用本工程的工程师，请熟读相关文档以及 update comment，并很希望大家能合作交流，多提 issues 和 pr，共同打造一个趁手好用的工作，多留出一些时间来摸鱼

CHAPTER 2

open-mmlab 介绍

> 因为本工程是在 openmmlab 的基础上做了一些小修改而搭建起来的，所以使用好本工程的前提是对 open-mmlab 的相关工作要有一定的了解，下面是相关简介

open-mmlab 中的系列框架是有依赖关系的，其本质是 OO(面向对象) 的继承表现

想快速了解 open-mmlab 一定要先了解工厂设计模式，在 open-mmlab 是通过 registey 来实现的，详情参考：registry 详解或者是对应的代码 (代码很短，推荐看一下)

简单介绍一下个人常用的其中几个工程

- mmcv：open-mmlab 中的基础库

  open-mmlab 其他各个框架中的基础功能都是来自于它或者继承自它

- mmdetection：2d 检测、分割的集成框架

  提供该领域里程碑式的模型支持，以及其他常见的口碑较好的模型支持，例如 RCNN 系列、yolo 系列等。更新速度一般晚于学术界半年左右

- mmdetection3d：3d 检测、分割的集成框架

  是在 mmdetection 基础上修改而来的，毕竟很多 3d 检测的方法用的其实还是 2d 的方法，更新速度一般晚于学术界半年左右，但是因为随着自动驾驶的火热以及 2D 赛道卷不动了大家纷纷转向 3D，所以其最近更新比较“迅猛”

- mmdeploy：一个部署框架

  其出现的目的是为了打通 open-mmlab 中项目落地的最后一步，其更新速度最慢，一般是一个已经非常成熟的模型才会被支持 (话说回来，如果支持速度很快，那大家都失业吧 ?) 各个被支持的模型一般支

持多种后端，最少是支持 onnx 和 tensorrt 目前还没有到 v1.0 的正式版对于被支持的模型，有的还会贴心的提供相关的 sdk，这样连前后处理都不需要自己写了 ~

CHAPTER 3

环境配置

为了后续使用的方便，最好按照如下需求准备好必要的 checkpoints、data、环境变量的配置

3.1 准备数据

准备训练、测试需要的数据、预训练的 checkpoints，并按照指定结构放置，用于后续配置数据软链接

3.1.1 checkpoints 数据

文件结构如下

```
mmlab_checkpoints
├── mmdetection
│   └── checkpoints
└── mmdetection3d
    └── checkpoints
```

3.1.2 data 数据

文件结构如下

```
mmlab_dataset
├── mmdetection
│   └── data
└── mmdetection3d
    └── data
```

3.2 拉取工程

为了减少数据量，一般仅拉取最新的 commit

```
git clone https://github.com/windzu/mmlab_extension.git --depth 1 && \
cd mmlab_extension && \
git submodule update --init --recursive
```

3.3 添加环境变量

根据自己使用的 shell 来配置，这里仅以 bash 为例

```
cd mmlab_extension && \
echo "export MMLAB_EXTENSION_PATH=$(pwd)" >> ~/.bashrc

cd mmlab_checkpoints && \
echo "export MMLAB_CHECKPOINTS_PATH=$(pwd)" >> ~/.bashrc

cd mmlab_dataset && \
echo "export MMLAB_DATASET_PATH=$(pwd)" >> ~/.bashrc
```

3.4 配置数据软链接

```
# checkpoints
rm -rf $MMLAB_EXTENSION_PATH/mmdetection/checkpoints && \
ln -s $MMLAB_CHECKPOINTS_PATH/mmdetection/checkpoints $MMLAB_EXTENSION_PATH/
↪mmdetection/checkpoints && \
rm -rf $MMLAB_EXTENSION_PATH/mmdetection3d/checkpoints && \
ln -s $MMLAB_CHECKPOINTS_PATH/mmdetection3d/checkpoints $MMLAB_EXTENSION_PATH/
```

(续下页)

(接上页)

```
↪mmdetection3d/checkpoints

# dataset
rm -rf $MMLAB_EXTENSION_PATH/mmdetection/data  && \
ln -s $MMLAB_DATASET_PATH/mmdetection/data $MMLAB_EXTENSION_PATH/mmdetection/data && \
rm -rf $MMLAB_EXTENSION_PATH/mmdetection3d/data  && \
ln -s $MMLAB_DATASET_PATH/mmdetection3d/data $MMLAB_EXTENSION_PATH/mmdetection3d/data
```

CHAPTER 4

Installation

目前仅提提供 conda 的安装方式

其中 mmcv 中的 cuda 算子编译依赖宿主机安装的 cuda, 所以在不同 cuda 版本下安装略有区别, 下面分别提供 ubuntu18.04+cuda10.2 和 ubuntu20.04+cuda11.3 的示例

4.1 ubuntu18.04 cuda10.2

```
# 待补充
```

4.2 ubuntu20.04 cuda11.3

```
export CONDA_ENV_NAME=mmdet && \
export PYTHON_VERSION=3.8 && \
export CUDA_VERSION=11.3 && \
export MMCV_VERSION=1.6.0 && \
export MMCV_CUDA_VERSION=113 && \
export TORCH_VERSION=1.12.0 && \
conda create -n $CONDA_ENV_NAME python=$PYTHON_VERSION -y && \
conda activate $CONDA_ENV_NAME && \
conda install pytorch=$TORCH_VERSION torchvision torchaudio cudatoolkit=$CUDA_VERSION␣
↪-c pytorch -y && \
```

(续下页)

(接上页)

```
pip install openmim && \
mim install mmcv-full==$MMCV_VERSION && \
cd $MMLAB_EXTENSION_PATH/mmdetection && \
pip install -e . && \
cd $MMLAB_EXTENSION_PATH/mmdetection_extension && \
pip install -e .
```

CHAPTER 5

Installation

目前仅提提供 conda 的安装方式

其中 mmcv 中的 cuda 算子编译依赖宿主机安装的 cuda, 所以在不同 cuda 版本下安装略有区别, 下面分别提供 ubuntu18.04+cuda10.2 和 ubuntu20.04+cuda11.3 的示例

5.1 ubuntu18.04 cuda10.2

```
export CONDA_ENV_NAME=mmdet3d && \
export PYTHON_VERSION=3.8 && \
export TORCH_VERSION=1.12.0 && \
export CUDA_VERSION=10.2 && \
export MMCV_CUDA_VERSION=102 && \
export MMCV_VERSION=1.6.0 && \
export MMDET_VERSION=2.25.0 && \
conda create -n $CONDA_ENV_NAME python=$PYTHON_VERSION -y && \
conda activate $CONDA_ENV_NAME && \
conda install pytorch=$TORCH_VERSION torchvision torchaudio cudatoolkit=$CUDA_VERSION␣
↪-c pytorch -y && \
cd $MMLAB_EXTENSION_PATH/mmcv && git checkout v$MMCV_VERSION && \
MMCV_WITH_OPS=1 pip install -e . && \
cd $MMLAB_EXTENSION_PATH/mmdetection && git checkout v$MMDET_VERSION && \
pip install -e . && \
pip install openmim && \
```

(续下页)

(接上页)

```
mim install mmsegmentation && \
cd $MMLAB_EXTENSION_PATH/mmdetection3d && \
pip install -e . && \
cd $MMLAB_EXTENSION_PATH/mmdetection3d_extension && \
pip install -e . && \
python3 -m pip install --user git+https://gitee.com/windzu/pypcd.git
```

5.2 ubuntu20.04 cuda11.3

```
export CONDA_ENV_NAME=mmdet3d && \
export PYTHON_VERSION=3.8 && \
export CUDA_VERSION=11.3 && \
export MMCV_VERSION=1.6.0 && \
export MMDET_VERSION=2.25.0 && \
export TORCH_VERSION=1.12.0 && \
conda create -n $CONDA_ENV_NAME python=$PYTHON_VERSION -y && \
conda activate $CONDA_ENV_NAME && \
conda install pytorch=$TORCH_VERSION torchvision torchaudio cudatoolkit=$CUDA_VERSION␣
↪-c pytorch -y && \
cd $MMLAB_EXTENSION_PATH/mmcv && git checkout v$MMCV_VERSION && \
MMCV_WITH_OPS=1 pip install -e . && \
cd $MMLAB_EXTENSION_PATH/mmdetection && git checkout v$MMDET_VERSION && \
pip install -e . && \
pip install openmim && \
mim install mmsegmentation && \
cd $MMLAB_EXTENSION_PATH/mmdetection3d && \
pip install -e . && \
cd $MMLAB_EXTENSION_PATH/mmdetection3d_extension && \
pip install -e . && \
python3 -m pip install --user git+https://gitee.com/windzu/pypcd.git
```

CHAPTER 6

Prerequisites

aaa

CHAPTER 7

Installation

只推荐 docker 安装

7.1 构建镜像

```
export MMDEPLOY_VERSION=0.7.0 && \
cd $MMLAB_EXTENSION_PATH/mmdeploy_extension && \
docker build docker/dev/ -t mmdeploy:$MMDEPLOY_VERSION \
--build-arg VERSION=$MMDEPLOY_VERSION \
--build-arg USE_SRC_INSIDE=true
```

7.2 创建容器

```
cd $MMLAB_EXTENSION_PATH/mmdeploy_extension && \
docker-compose up -d
```