10/30光轮for吉利交付文档

光轮交付项：

针对上轮交付吉利提出的问题以及改进意见，光轮本次交付如下内容：

- 训练/测评的多SQLite DB支持
- DeepSpeed多机多卡训练支持
- 导航模块的设计方案

加载最新版交付docker：

Docker 下载链接：train_2.1.1.tar.bz2

```
# sudo apt install lbzip2
cat train_2.1.1.tar.bz | pv | lbzip2 -d | docker load
```

1. 多 SQLite Database Dataloader支持

本次交付中，光轮快系统支持在训练中同时读取多个 `.db` 文件。

- 使用方法：

在 `lwad/adbase/vad/configs/VAD/VAD_lightwheel_config.py` 的 `18` 行附近。

```
ann_file_train = "/data/ceph/data/nuplan/ann_files/ann_dir/"
```

`ann_file_train` 参数现在支持一个文件夹，内部是多个 `.db` 的sqlite文件。

- 训练命令行

在 `/workspace/lwad/lwad` 目录下：

```
torchrun \
    --nproc_per_node=4 \
    --master_port=28509 \
    ./adbase/vad/train.py \
    ./adbase/vad/configs/VAD/VAD_lightwheel_config.py \
    --launcher=pytorch \
```

```
    --deterministic
```

2. DeepSpeed 支持多机多卡训练

本次交付文档展示docker容器下的DeepSpeed多机训练。

DeepSpeed API是PyTorch的一个轻量级封装，可以实现分布式的多机多卡训练。多机使用docker容器部署环境，具体请参考上次交付文档中关于docker训练的说明。

假设在服务器A和服务器B上进行多机训练，服务器A的IP为 `192.168.100.250` ，服务器B的IP为 `192.168.100.248` ，服务器A为多机训练的master主机。

2.1 docker容器环境配置

假设服务器A和服务器B上的docker容器名分别为 `lwad` 和 `lwad2` ：

在服务器A上：

```
docker run -it \
--name lwad \
--user root \
-P \
--shm-size 64G --ulimit memlock=-1 --ulimit stack=67108864 \
--runtime=nvidia  -e NVIDIA_VISIBLE_DEVICES=all -e
NVIDIA_DRIVER_CAPABILITIES=all --gpus all \
-v /data/ceph:/data/ceph \
harbor.lightwheel.net/lwad/train:3.1.0
```

在服务器B上：

```
docker run -it \
--name lwad2 \
--user root \
-P \
--shm-size 64G --ulimit memlock=-1 --ulimit stack=67108864 \
--runtime=nvidia  -e NVIDIA_VISIBLE_DEVICES=all -e
NVIDIA_DRIVER_CAPABILITIES=all --gpus all \
-v /data/ceph:/data/ceph \
harbor.lightwheel.net/lwad/train:3.1.0
```

在每台机器上使用 `docker port` 命令查看lwad容器对应的端口，并配置 `ds_config/hostfile.txt`

可以使用 `docker port` 命令在服务器A和B上分别查看 `lwad` 和 `lwad2` 的容器端口号。

```
# 在服务器A上查看lwad容器端口
docker port lwad

# 输出示例
22/tcp -> 0.0.0.0:32769
22/tcp -> [::]:32769
# 32769即表示lwad容器在服务器A上对应的端口号

# 在服务器B上查看lwad_2容器端口
docker port lwad2

# 输出示例
22/tcp -> 0.0.0.0:32768
22/tcp -> [::]:32768
# 32768即表示lwad2容器在服务器B上对应的端口号
```

则 `hostfile.txt` 配置如下，代表两台机器，每台机器上各使用一张GPU卡，lwad使用32769端口和GPU0和GPU1，lwad2使用32768端口和GPU0和GPU1。

```
192.168.100.250:32769 slots=2 gpu=0,1
192.168.100.248:32768 slots=2 gpu=0,1
```

2.2 验证服务器建立SSH通信

测试无密码连接是否成功：

```
ssh root@192.168.100.248 -p 32768
# 应该能够直接登录，无需输入密码
```

若无密码连接成功，即可建立服务器A到服务器B的SSH通信。

2.3 多机训练配置

接下来在项目主目录（`/workspace/lwad/lwad`）里运行以下命令：

```
deepspeed --hostfile deepspeed_cfg/hostfile.txt \
          --master_addr 192.168.100.250:32769 \
          --master_port 29600 \
          ./adbase/vad/train.py \
          ./adbase/vad/configs/VAD/VAD_lightwheel_config.py \
          --launcher pytorch \
          --deterministic \
          --deepspeed \
          --deepspeed_config ./ds_config/ds_config.json
```

在命令中指定`--deepspeed`和`--deepspeed_config`，分别指定使用deepspeed和deepspeed配置文件。

`deepspeed_config`是deepspeed的配置文件，以下为一个示例：

```
{
    "train_batch_size": 8,
    "train_micro_batch_size_per_gpu": 1,
    "steps_per_print": 2000
}
```

- `train_batch_size`指定训练的批量大小。
- `train_micro_batch_size_per_gpu`指定每个GPU卡上的mini batch大小。

需要注意的是，`train_batch_size`必须是`train_micro_batch_size_per_gpu`的整数倍。

即可成功跑起多机训练。

更多关于deepspeed配置文档的详解，请参考DeepSpeed官方配置文档。

3. 导航模块设计方案

3.1 现阶段command的优化

针对现阶段导航的局限性（例如可能混淆左右变道和左右转），光轮计划配合慢系统输出的meta action对导航命令进行细致的划分。

慢系统中，对指令的定义如下：

注：Meta action表示短期驾驶决策，如加速、减速、转弯或变道。

Category	Meta-actions
Speed-control actions	Speed up, Slow down, Slow down rapidly, Go straight slowly, Go straight at a constant speed, Stop, Wait, Reverse
Turning actions	Turn left, Turn right, Turn around
Lane-control actions	Change lane to the left, Change lane to the right, Shift slightly to the left, Shift slightly to the right

3.2 Raster 地图表征导航信息

3.2.1 导航信息获取

- 基于 nuPlan 的训练与闭环仿真：使用 OpenStreetMap 或谷歌地图，也可使用 A* 算法获取场景级别路径
- 实车数据采集：使用百度地图或内部系统， 需要吉利对齐车上使用的导航地图模态（图像/文本）。

3.2.2 Raster 地图表征

- 将主车周围 80m 左右的 BEV 范围划为单位宽度 0.25-0.5m 的网格（尺寸根据实际情况调整）
- 根据高精地图与导航信息，将导航路径经过的可行区域涂黑，如左转车道，用resnet等做特征提取。
- 基于导航的语义信息，如：200米后左转，制作 one-hot 编码 （计划借助蒸馏的语言模型实现，还要实验）

3.2.3 模型输入

- 通过 MLP 将导航信息表征转化为 Embedding
- 将导航 Embedding 作为 Planner 的一个输入

注：可对导航信息 Embedding 进行预训练后再并入模型共同训练