

目录

目录

一、实验原理及方案选型	3
(一) 环境配置	3
1. 虚拟环境	3
2. 用于 Linux 的 Windows 子系统	3
3. Linux 系统	4
(二) 端边云协同	4
1. 端-边-云	4
2. ModelArts 一站式开发平台	4
3. 自动学习	4
4. Workflow (工作流)	4
(三) 手写数字识别	5
二、实验过程	5
(一) 环境配置	5
1. WSL2 安装	5
2. 子系统换盘	6
3. 为子系统安装 Anaconda	8
4. VScode 远程连接	9
(二) 端边云协同	10
1. 准备工作	10
2. 自动学习基础	14
3. 开发环境实践	17
4. ModelArts Standard 自定义算法	19
(三) 手写数字识别	22
1. 下载 MindSpore	22
2. 代码准备	22
三、实验结果	23
(一) 环境配置	23
1. wsl2 安装	23

2. Ubuntu 换盘	23
3. python 环境配置	24
(二) 端边云协同	24
1. 自动学习	24
2. 开发环境实践	25
3. ModelArts Standard 自定义算法	25
(三) 手写数字识别	26
四、实验存在的问题	26
(一) 环境配置	26
(二) 端边云协同	26
ModelArts Standard 自定义算法	26
(三) 手写数字识别	27
五、心得体会及建议	28
1. WSL2 与 Anaconda 环境的配置	28
2. 端边云协同的学习与实践	28
3. MindSpore 框架的本地训练	28
4. 总结与展望	28
5. 建议	28

一、实验原理及方案选型

(一) 环境配置

1. 虚拟环境

虚拟机（通常简称为 VM）：包括用于存储文件的 CPU、内存和磁盘，并可连接到 Internet。构成计算机的部件是物理的、有形的。VM 通常被认为是物理服务器中的虚拟计算机或软件定义的计算机，仅以代码的形式存在。

虚拟化是创建基于软件或计算机的“虚拟”版本的过程。其包含从物理主机计算（如个人计算机）和/或远程服务器（如云提供商的数据中心的服务器）“借用”的专用 CPU、内存和存储量。

2. 用于 Linux 的 Windows 子系统

WSL 2 是适用于 Linux 的 Windows 子系统体系结构的一个新版本，它支持适用于 Linux 的 Windows 子系统在 Windows 上运行 ELF64 Linux 二进制文件。它的主要目标是提高文件系统性能，以及添加完全的系统调用兼容性。

3. Linux 系统

Linux，全称 GNU/Linux，是一套免费使用和自由传播的类 Unix 操作系统，是一个基于 POSIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。

Linux 系统分为内核、shell、文件系统和应用程序。内核、shell 和文件系统一起形成了基本的操作系统结构，它们使得用户可以运行程序、管理文件并使用系统。

Linux 内核由如下几部分组成：内存管理、进程管理、设备驱动程序、文件系统和网络管理等。

（二）端边云协同

1. 端-边-云

分别指端侧设备、智能边缘设备、公有云。

2. ModelArts 一站式开发平台

一站式开发平台指 AI 开发的各个环节，包括数据处理、算法开发、模型训练、模型部署都可以在该平台上完成

ModelArts 是面向 AI 开发者的一站式开发平台，提供海量数据预处理及半自动化标注、规模分布式训练、自动化模型生成及端-边-云模型按需部署能力，帮助用户快速创建和部署模型，管理全周期 AI 工作流。

3. 自动学习

自动学习功能可以根据标注数据自动设计模型、自动调参、自动训练、自动压缩和部署模型，不需要代码编写和模型开发经验。只需三步，标注数据、自动训练、部署模型，即可完成模型构建。

4. Workflow（工作流）

Workflow 本质是开发者基于实际业务场景开发用于部署模型或应用的流水线工具。在机器学习的场景中，流水线可能会覆盖数据标注、数据处理、模型开发/训练、模型评估、应用开发、应用评估等步骤。

区别于传统的机器学习模型构建，开发者可以使用 Workflow 开发生产流水线。基于 MLOps 的概念，Workflow 会提供运行记录、监控、持续运行等功能。根据角色的分工与概念，产品上将工作流的开发和持续迭代分开。

总的来说，Workflow 有两种形态：

- 开发态：使用 Workflow 的 Python SDK 开发和调测流水线。
- 运行态：可视化配置运行生产好的流水线。

（三）手写数字识别

LeNet 模块主要有模块组成，前 2 个是卷积模块，后 1 个是全连接模块卷积模块中主要模块是卷积层与池化层

卷积层中都使用 5X5 的卷积核，并使用 sigmoid 的激活函数，第一层卷积输出通道为 6，第二层卷积输出通道是 16，在每一层卷积中的池化层选择最大池化层，最大池化层选取 2X2 的窗口，并且步长为 2，由于池化窗口与步伐相同，所以不重叠，C5 中输出为 120*1*1。

全连接层 F6 中，输入向量维度：[1, 120]，权重参数的维度：[120, 84]，得到一个输出维度[1, 86]的一维向量，最后经过 output 层输出维度[1, 10]，10 为输出的类别个数。

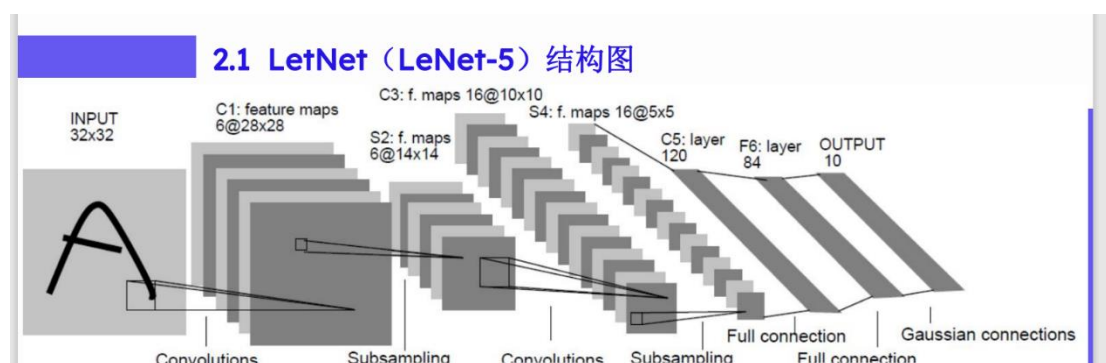


图 1

二、实验过程

（一）环境配置

1. WSL2 安装

①Windows11 家庭版没有自带的 Hyper-V 需要自己安装，在网上查阅了教程，将一段代码（如图 2）保存为 hyper-v.bat 文件并运行，便可以在控制面板的程序中找到 Hyper-V 功能（如图 3）。

```
pushd "%~dp0"
dir /b %SystemRoot%\servicing\Packages\*Hyper-V*.mum >hyper-v.txt
for /f %i in ('findstr /i . hyper-v.txt 2^>nul') do dism /online /norestart /add-pack
del hyper-v.txt
Dism /online /enable-feature /featurename:Microsoft-Hyper-V-All /LimitAccess /ALL
```

图 2

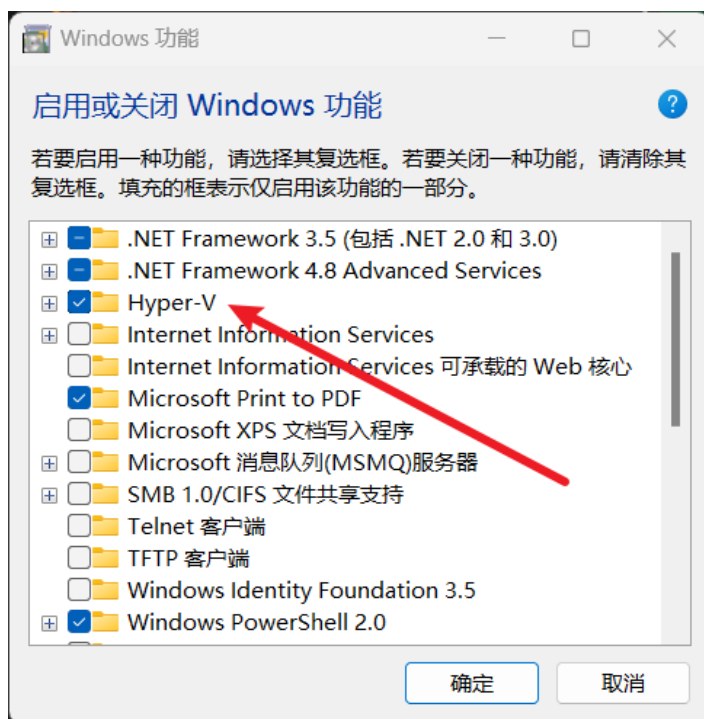


图 3

②控制面板->程序->启用或关闭 windows 功能，开启 Windows 虚拟化和 Linux 子系统（WSL2）以及 Hyper-V。勾选完成后，Windows11 会下载些东西，并提示重启。等电脑彻底重启完以后，进行后续操作

③打开命令提示符或者 Power Shell，运行 `wsl --install` 命令，会默认下载最新版本的 Ubuntu（如图 4），如果需要特定版本可以在命令后指定，如 `wsl --install -d Ubuntu-20.04`。

```
PS C:\Windows\system32> wsl --install
正在安装: Ubuntu
已安装 Ubuntu。
正在启动 Ubuntu...
Installing, this may take a few minutes...
```

图 4

④之后通过命令行打开 Ubuntu 系统，根据系统的提示设置用户名和密码，即可开始使用。

2. 子系统换盘

为了防止占用系统盘太多空间，我们建议将子系统换盘（我将其换到了 D 盘）（如图 5）。

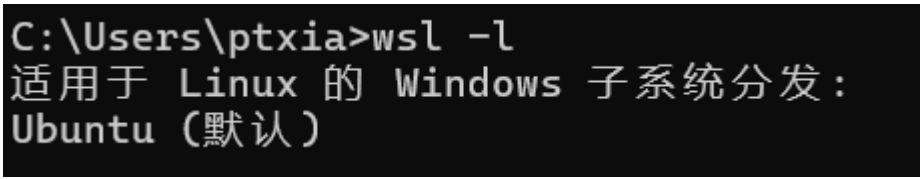


图 5

①导出发行版本（注意尽量不要改变发行版本名称）：

wsl --export <发行版名称> <绝对\路径\文件名.tar>

注意路径中不要有空格，如 Program Files。否则该命令无法运行。

②卸载发行版本：wsl--unregister <发行版名称>

在执行此命令之前一定要确定导出的位置已经能看到<xx.tar>文件，否则说明还未导出（如图 6）。

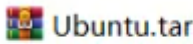


图 6

③导入发行版本 :wsl-import <发行版名称> <导入后\存放\绝对路径\目录> <要导入的\绝对\路径\文件名.tar>

导入的结果为一个硬盘映像文件（如图 7），无法通过鼠标点击打开。

ext4.vhdx	2024/7/20 16:26	硬盘映像文件	11,820,03...
-----------	-----------------	--------	--------------

图 7

④修改用户名，换盘之后默认用 root 用户登录，之前的设置会消失，所以需要修改用户。Win+R 输入 regedit 打开注册表编辑器，找到 HKEY_USERS\xxx\Software\Microsoft\Windows\CurrentVersion\Lxss（如图 8），修改其子文件夹下的 DefaultUid 为 3e8（十六进制）或 1000（十进制）（如图 9），root 用户为 0。之后默认用户即改为先前设置的用户。

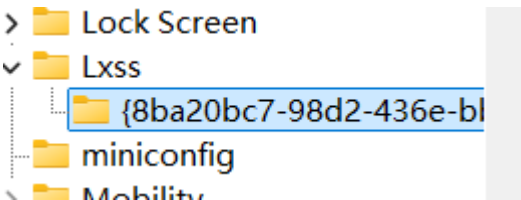


图 8

(默认)	REG_SZ	(数值未设置)
BasePath	REG_SZ	\\?\D:\wsl
DefaultUid	REG_DWORD	0x000003e8 (1000)
DistributionNa...	REG_SZ	Ubuntu
Flags	REG_DWORD	0x0000000f (15)
State	REG_DWORD	0x00000001 (1)
Version	REG_DWORD	0x00000002 (2)

图 9

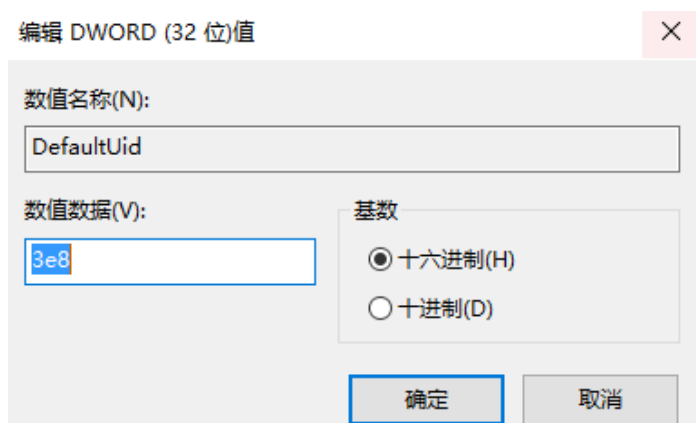


图 10

3. 为子系统安装 Anaconda

①首先需要更新一下 apt-get，否则后续会报错

```
xumian@Xumian:~$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu jam
Get:2 http://security.ubuntu.com/ubuntu ja
Get:3 http://archive.ubuntu.com/ubuntu jam
```

图 11

②在 Ubuntu18.04 发行版安装 SSH 工具和 wget 工具。

```
sudo apt-get install openssh-server
```

```
apt-get install -y wget
```

③下载 anaconda，并修改下载文件的权限为可执行

```
cd /home/xxx(此处为 user 目录)
```

```
wget <anaconda 下载网址>
```

```
chmod +x Anaconda3-2023.03-1-Linux-x86_64.sh
```

④安装 `./Anaconda3-2023.03-1-Linux-x86_64.sh`

之后一直按 enter 键，若需要确定条款或安装位置就输入 yes，最后询问是否安装 vscode，选择 no，即安装完成，通过 `conda -V` 检查（如图 12）

```
(base) xumian@Xumian:~$ conda -V
conda 23.3.1
(base) xumian@Xumian:~$ |
```

图 12

4. VScode 远程连接

VScode 打开 remote explorer, 选择 WSL Targets 可以看到本地安装的 Ubuntu (如图 13), 点击连接。

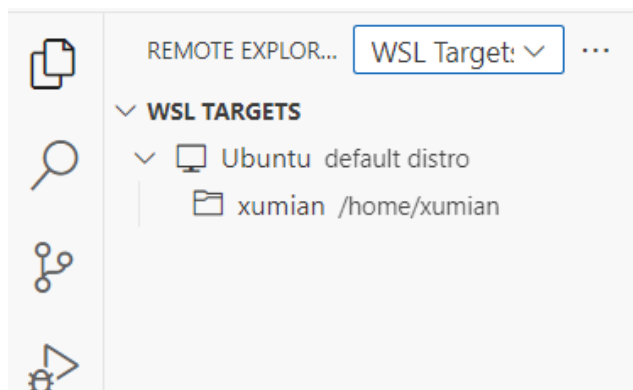


图 13

安装相关插件, 之前在 windows 上已安装过的插件也需要重新安装。(图 14)

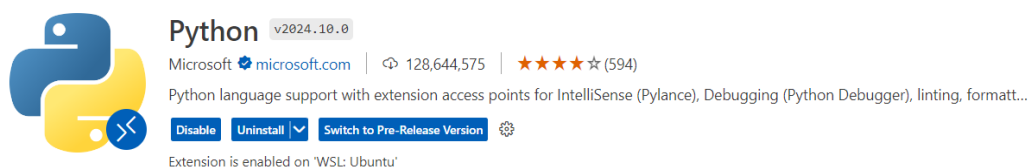


图 14

创建 conda 环境 (图 15)

```
(base) xumian@Xumian:~$ conda env list
# conda environments:
#
base                  *  /home/xumian/anaconda3
zn                    /home/xumian/anaconda3
```

图 15

新建 python 文件, 并选择虚拟环境, 编写 koch.py 文件 (如图 16)
科赫曲线的基本概念和绘制方法如下:

- 正整数 n 代表科赫曲线的阶数, 表示生成科赫曲线过程的操作次数。
- 科赫曲线初始化阶数为 0, 表示一个长度为 L 的直线。
- 对于直线 L , 将其等分为三段, 中间一段用边长为 $L/3$ 的等边三角形的两个边替代, 得到 1 阶科赫曲线, 它包含四条线段。
- 进一步对每条线段重复同样的操作后得到 2 阶科赫曲线。继续重复同

样的操作 n 次可以得到 n 阶科赫曲线

```
import turtle
def koch(size,n):
    if n==0:
        turtle.fd(size)
    else:
        for angle in[0,60,-120,60]:
            turtle.left(angle)
            koch(size/3,n-1)
def main():
    turtle.setup(800,400)
    turtle.speed(0)#控制绘制速度
    turtle.penup()
    turtle.goto(-300,-50)
    turtle.pendown()
    turtle.pensize(2)
    koch(600,6)#0阶科赫曲线长度, 阶数
    turtle.hideturtle()
main()
```

图 16

运行结果:

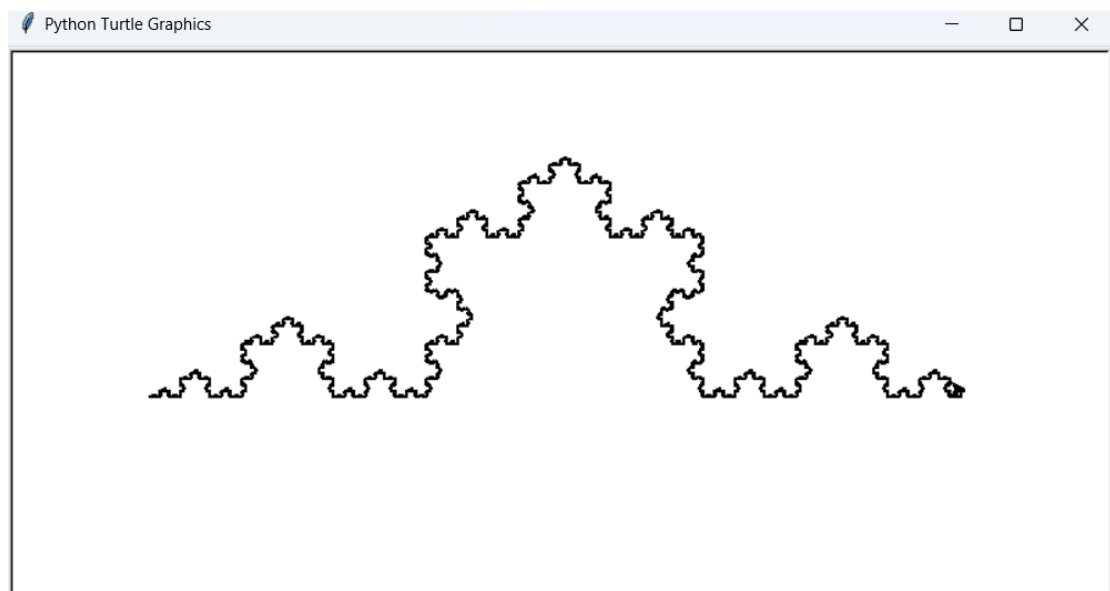


图 17

(二) 端边云协同

1. 准备工作

注册华为云账号（<https://www.huaweicloud.com/>），并进行实名认证（个人认证）。配置访问授权（全局配置），登录 ModelArts 管理控制台，在左侧导航栏选择“全局配置”，进入“全局配置”页面（注意选择：华北-北京四）。

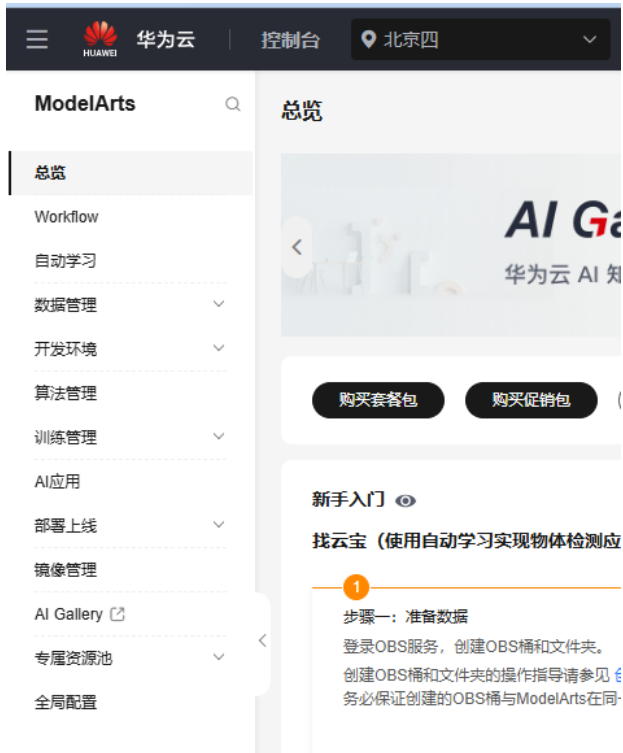


图 18

创建 OBS 桶，*登陆对象存储服务-控制台(huaweicloud.com)，*在桶列表页面右上角单击“创建桶”，创建 OBS 桶，创建名称为“xxxx-xxxx”的 OBS 桶



图 19

创建桶后需要购买资源包



图 20

存储包 流量包



图 21

购买 ECS 云服务器，根据需要选择 CPU 型或 GPU 加速型，（参数如下）

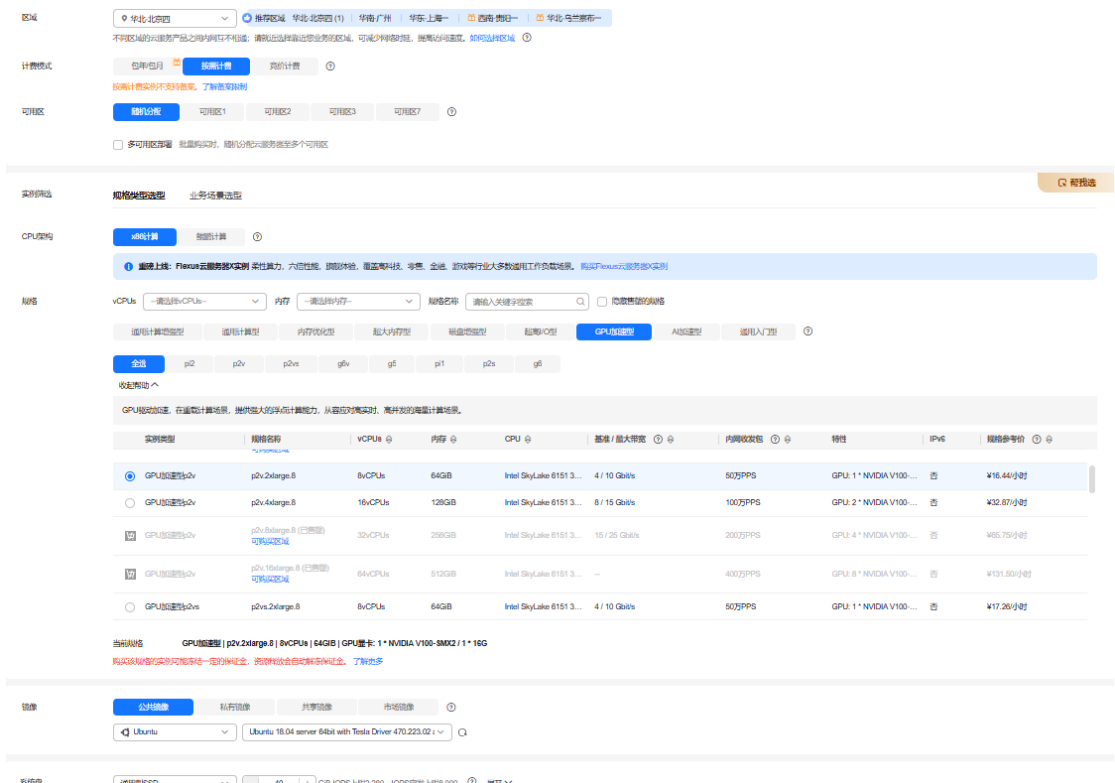


图 22

配置安全组

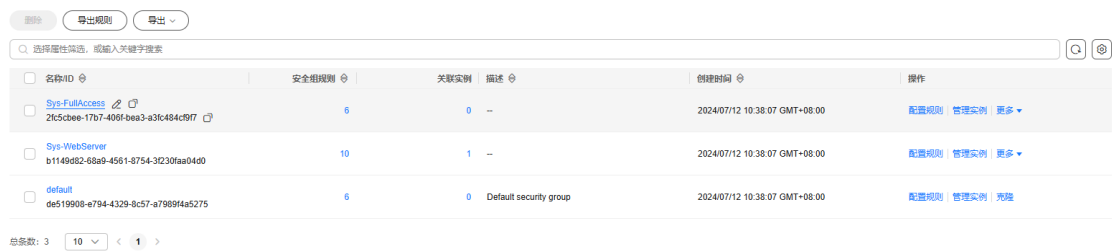


图 23

配置弹性公网 IP



图 24

安装 OBSBrowser+

https://support.huaweicloud.com/browsertg-obs/obs_03_1003.html

登录：OBSBrowser+

https://support.huaweicloud.com/browsertg-obs/obs_03_1004.html

AK方式登录

账号登录

授权码登录

账号名 ?

ning

服务提供商 ?

华为对象存储服务（默认）

Access Key ID

图 25

2. 自动学习基础

A 口罩检测

(1) 在 OBS 桶中创建文件夹用于存储存储数据集

☐ 名称

☐ input

☐ output

图 26

下载数据集到 OBS 桶中

<input type="checkbox"/> 名称	存储类别	大小 令
<input type="checkbox"/> b001_1580797659815.jpg	标准存储	37.77 KB
<input type="checkbox"/> b001_1580797659815.xml	标准存储	1.33 KB
<input type="checkbox"/> b002_1580797660107.jpg	标准存储	31.78 KB
<input type="checkbox"/> b002_1580797660107.xml	标准存储	1,022 bytes
<input type="checkbox"/> b003_1580797660133.jpg	标准存储	33.81 KB
<input type="checkbox"/> b003_1580797660133.xml	标准存储	685 bytes

图 27

(2) 创建自动学习物体检测项目

在 ModelArts 控制台，左侧导航栏选择“自动学习”默认进入新版自动学习页面，

选择物体检测项目，单击“创建项目”。

★ 计费模式

按需计费

★ 名称

mask_detect

描述

口罩检测

4/500

★ 数据集

dataset-b4f1

创建数据集

★ 输出路径

/zn11-mask/mask/output/

选择

★ 训练规格

[限时免费]GPU: 1*GP-Vnt1(32GB) | CPU: 8 ...

图 28

运行数据标注

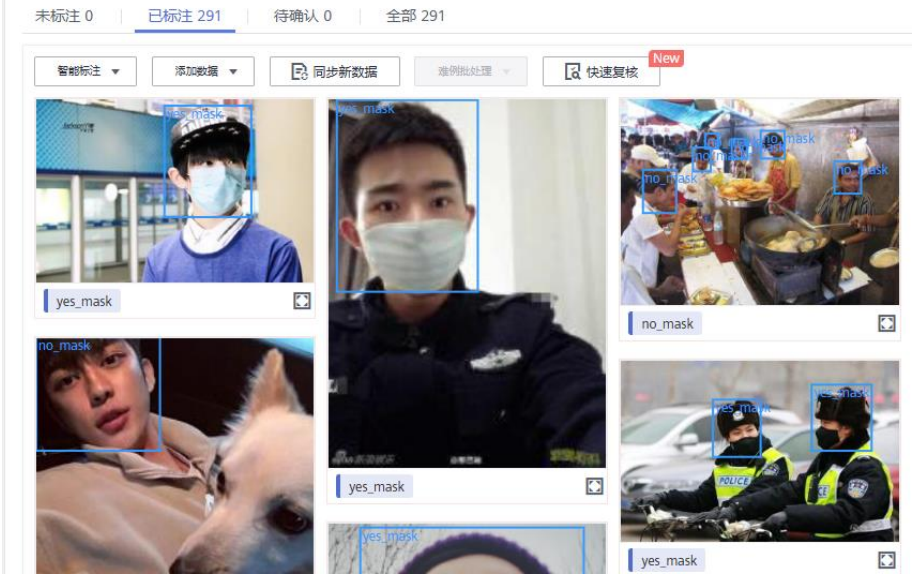


图 29

(3) 运行 workflow



图 30

运行到服务部署节点需要设置两个参数
计算节点规格：根据您的实际需求选择相应的规格。

是否自动停止：为避免资源浪费，建议打开自动停止开关，根据您的实际需要，选择自动停止时间，也可以自定义自动停止的时间。

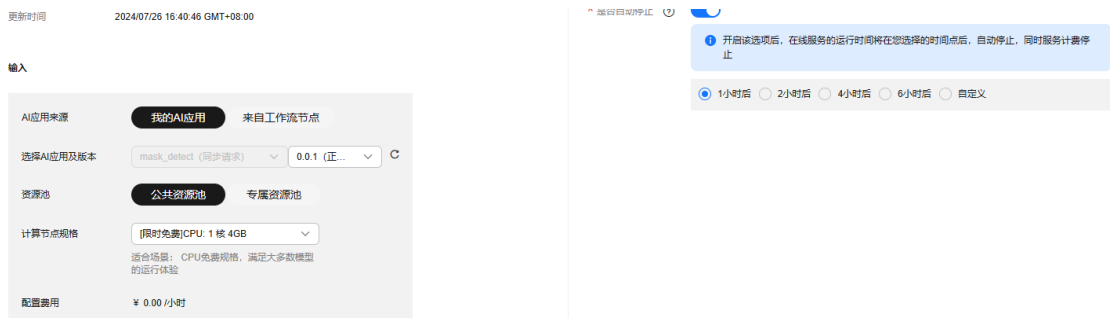


图 31

(4) 预测分析

在服务部署节点单击“实例详情”直接跳转进入在线服务详情页，或者在ModelArts 管理控制台，选择“部署上线>在线服务”，单击生成的在线服务名称，即可进入在线服务详情页。在服务详情页，选择“预测”页签。上传预测照片。

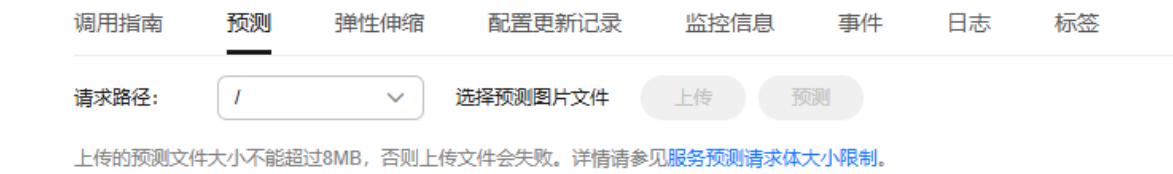


图 32

查看预测结果

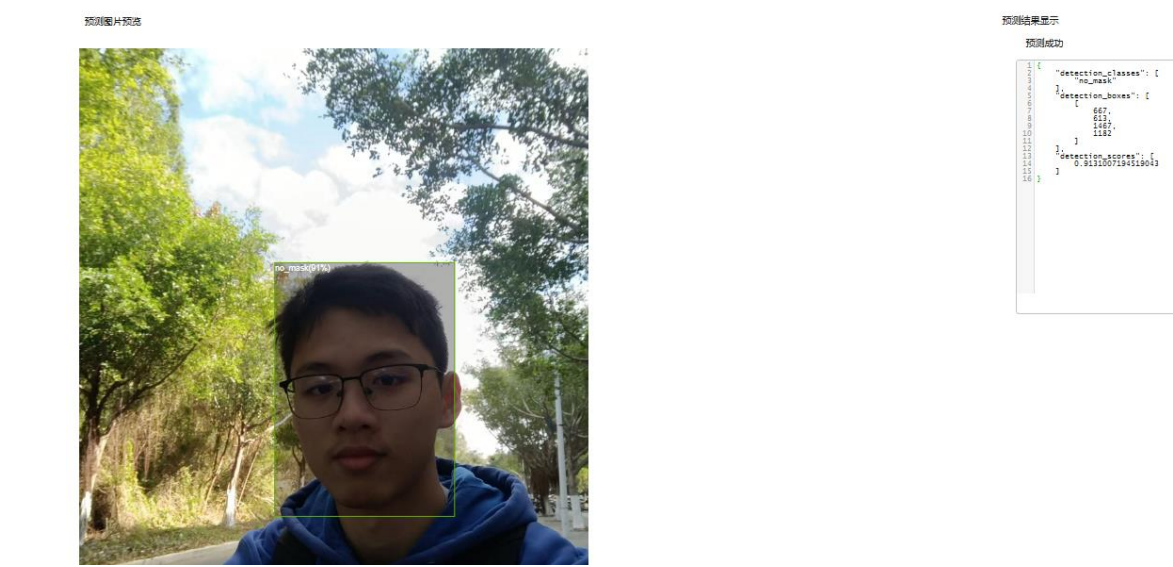


图 33

预测成功

(5) 清除相应的资源

停止运行服务，删除 OBS 对象

B 垃圾分类

仿照上一步下载数据

8类常见生活垃圾图片数据集

下载中38%

发布者 WAYNE

2024-07-27 13:32

文件大小 20.27MB

文件数量 1600

图 34

运行 workflow，标注数据，部署模型


未标注 0 | 已标注 800 | 待确认 0 | 全部 800

管理标注


添加数据

同步新数据


删除批处理




其他垃圾_一次性餐盒




可回收物_纸板箱




有害垃圾_过期药物




其他垃圾_烟蒂




有害垃圾_过期药物



可回收物_纸板箱



厨余垃圾_蛋壳

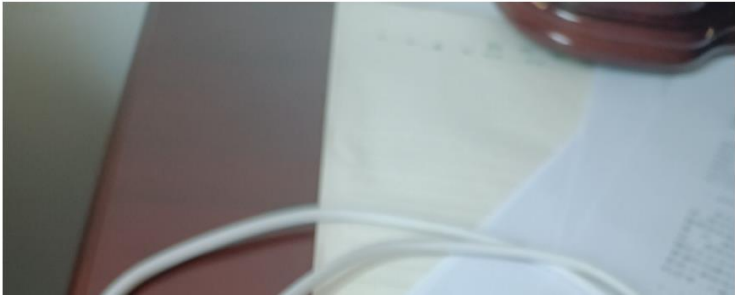


有害垃圾_过期药物

图 35

测试

预测图片预览



预测结果显示

预测成功

```
1 {
2   "predicted_label": "有害垃圾_过期药物",
3   "scores": [
4     [
5       "有害垃圾_过期药物",
6       "0.516"
7     ],
8     [
9       "其他垃圾_一次性餐盒",
10      "0.445"
11     ],
12     [
13       "可回收物_塑料玩具",
14       "0.035"
15     ],
16     [
17       "厨余垃圾_蛋壳",
18       "0.002"
19     ],
20     [
21       "其他垃圾_烟蒂",
22       "0.001"
23     ]
24   ]
25 }
```

图 36

删除资源

3. 开发环境实践

(1) 创建 Notebook 实例



图 37

(2) 利用 Jupyterlab 管理

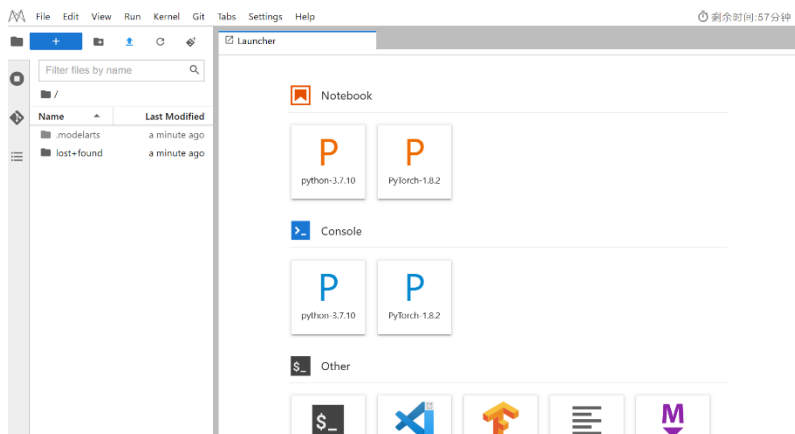


图 38

(3) 尝试使用 CodeLab (提供一部分时间的免费算力)

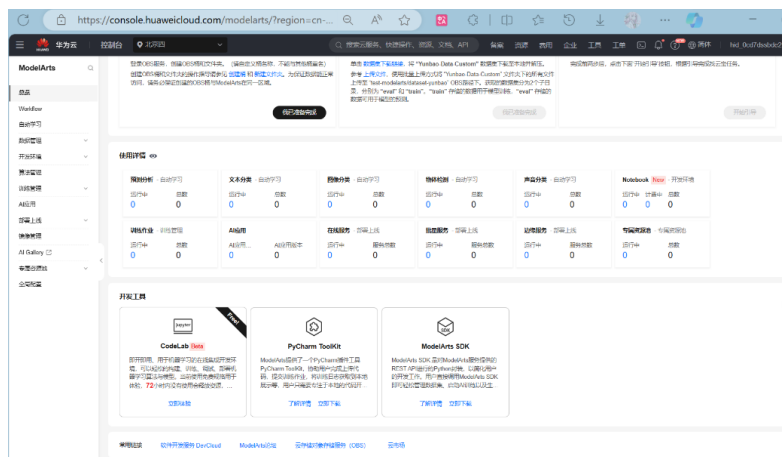


图 39



图 40

(4) VScode 上安装插件

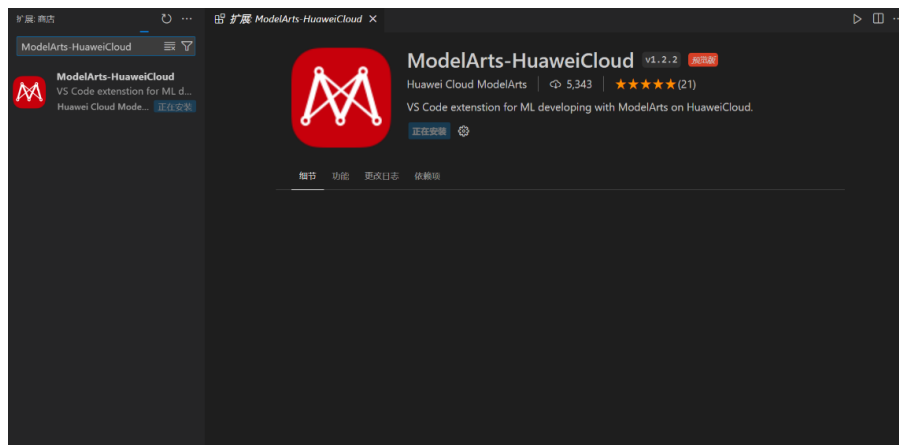


图 41

(5) 连接本地 VScode

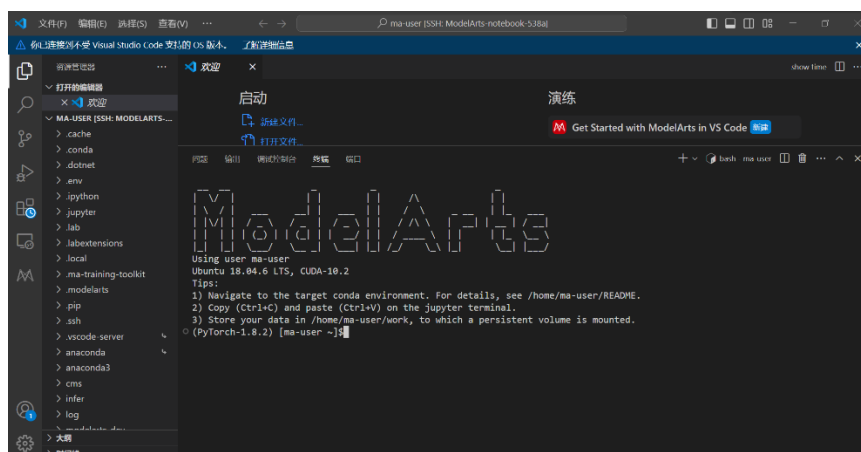


图 42

4. ModelArts Standard 自定义算法

(1) 在本地准备好 train.py, customize_service.py, config.json 文件（详见附件）

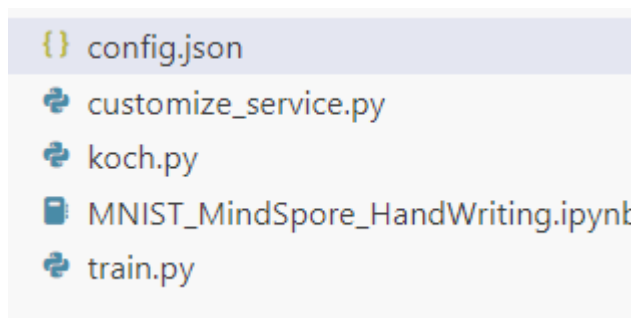


图 43

(2) 创建桶与文件夹，并上传对应文件与数据



图 44

(3) 创建训练作业

在左侧导航栏选择“训练管理 > 训练作业”进入训练作业页面，单击“创建训练作业”。选择“自定义算法”



图 45

(4) 模型训练完成后，可以创建 AI 应用，将 AI 应用部署为在线服务。

在 ModelArts 管理控制台，单击左侧导航栏中的“AI 应用管理>AI 应用”，进入“我的 AI 应用”页面，单击“创建”

★ 元模型来源 ?

从训练中选择 从对象存储服务 (OBS) 中选择 从容器镜像中选择

1. 导入ModelArts训练作业中训练完成的模型。请在下方选择需要导入的训练作业。模型规范详情参见 [从训练](#)
2. 使用容器化部署，导入的元模型有大小限制，详情请参见 [导入AI应用对于镜像大小限制](#)。

★ 选择训练作业 job-465f-copy-d1f0 ▼

☒ 动态加载 ?

★ AI引擎 PyTorch ▼ pytorch_1.8.0-cuda_10.2-... ▼

如果您的模型文件和推理代码没有在本地上运行过，可以尝试在开发环境调试后导入AI应用。请参考 [无感知迁移](#)

图 46

(5) 部署在线服务，如果使用限时免费节点，需要删除先前已经部署的使用限时免费节点的服务。

<input type="checkbox"/>	名称/ID 🔍	状态
<input type="checkbox"/>	service-935c 5ded90fc-ab24-4c9c-b639-458909ac5528	免费 → 运行中 (58 分钟后停止)

图 47

(6) 预测结果

单击“预测”页签，请求类型选择“multipart/form-data”，请求参数填写“image”，单击“上传”按钮上传示例图片，然后单击“预测”。

请求路径: 请求类型: multipart/form-data ▼ ★ 请求参数: image 选择预测文件 上传 重新预测

上传的预测文件大小不能超过8MB，否则上传文件会失败。详情请参见[服务预测请求体大小限制](#)。

预测图片预览



预测结果显示

预测成功

```
[{"image": 2}]
```

图 48

能够成功预测

(三) 手写数字识别

1. 下载 MindSpore

(1) Windows 版本:

```
pip install
https://ms-release.obs.cn-north-4.myhuaweicloud.com/1.1.0/MindSpore/c
pu/windows_x64/mindspore-1.1.0-cp37-cp37m-win_amd64.whl
--trusted-host ms-release.obs.cn-north-4.myhuaweicloud.com -i
https://pypi.tuna.tsinghua.edu.cn/simple
conda install mindspore-cpu=2.0.0 -c mindspore -c conda-forge
```

mkl-2024.1.0	104.3 MB	#####7	100%
mindspore-cpu-2.0.0	63.9 MB	#####	100%
pillow-9.2.0	45.4 MB	#####	100%
scipy-1.7.3	24.4 MB	#####	100%
python-3.7.12	18.0 MB	#####	100%
numpy-1.21.6	5.3 MB	#####	100%
libblas-3.9.0	5.0 MB	#####	100%
liblapack-3.9.0	5.0 MB	#####	100%

图 49

(2) Ubuntu 版本

```
pip install
https://ms-release.obs.cn-north-4.myhuaweicloud.com/1.1.0/MindSpore/c
pu/ubuntu_x86/mindspore-1.1.0-cp37-cp37m-linux_x86_64.whl
--trusted-host ms-release.obs.cn-north-4.myhuaweicloud.com -i
https://pypi.tuna.tsinghua.edu.cn/simple
```

2. 代码准备

代码来源

https://www.mindspore.cn/tutorials/zh-CN/rl.6/quick_start.html

在本地选择先前创建的虚拟环境，并运行代码

```
epoch: 1 step: 125, loss is 2.3006350994110107
epoch: 1 step: 250, loss is 2.323580503463745
epoch: 1 step: 375, loss is 2.2982375621795654
epoch: 1 step: 500, loss is 2.3163111209869385
epoch: 1 step: 625, loss is 2.3026835918426514
epoch: 1 step: 750, loss is 2.3133859634399414
epoch: 1 step: 875, loss is 2.289008378982544
epoch: 1 step: 1000, loss is 2.2915260791778564
epoch: 1 step: 1125, loss is 0.8956296443939209
epoch: 1 step: 1250, loss is 0.3606884777545929
epoch: 1 step: 1375, loss is 0.22799231112003326
epoch: 1 step: 1500, loss is 0.053907472640275955
epoch: 1 step: 1625, loss is 0.062161270529031754
epoch: 1 step: 1750, loss is 0.04021862521767616
```

图 50

三、实验结果

（一）环境配置

1. ws12 安装

成功安装适用于 Linux 的 Windows 子系统



图 51

2. Ubuntu 换盘

成功安装将 Ubuntu22.04 换到非系统盘，并修改用户（非 root 用户），能够通过命令提示符启动。

```

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/xumian/.hushlogin file.
(base) xumian@Xumian:~$ |

```

图 52

3. python 环境配置

成功安装 Linux 系统的 Anaconda，并创建虚拟环境

```

(base) xumian@Xumian:~$ conda env list
# conda environments:
#
base                  *  /home/xumian/anaconda3
zn                    /home/xumian/anaconda3/envs/zn

```

图 53

```

lz4-c                 1.9.4                h6a678d5_0
lzo                   2.10                 h7b6447c_2
markdown              3.4.1                py310h06a4308_0
markupsafe            2.1.1                py310h7f8727e_0
matplotlib            3.7.0                py310h06a4308_0
matplotlib-base       3.7.0                py310h1128e8f_0
matplotlib-inline     0.1.6                py310h06a4308_0
mccabe                0.7.0                pyhd3eb1b0_0
mistune               0.8.4                py310h7f8727e_1000
mkl                   2021.4.0             h06a4308_640
mkl-service           2.4.0                py310h7f8727e_0
mkl_fft               1.3.1                py310hd6ae3a3_0
mkl_random             1.2.2                py310h00e6091_0

```

图 54

(二) 端边云协同

1. 自动学习

口罩检测模型成功部署，并能够对上传的照片进行预测

事件类型	事件信息
✓ 正常	Image built successfully.
✓ 正常	The status of the image building task is READY.
✓ 正常	The status of the image building task is CREATING.
✓ 正常	The status of the image building task is CREATING.
✓ 正常	The status of the image building task is CREATING.
✓ 正常	The status of the image building task is CREATING.
✓ 正常	The status of the image building task is CREATING.
✓ 正常	The status of the image building task is CREATING.
✓ 正常	The status of the image building task is CREATING.
✓ 正常	The status of the image building task is CREATING.

图 55

垃圾分类模型成功部署

预测图片预览



预测结果显示

预测成功

```
1 {
2   "predicted_label": "其他垃圾_烟蒂",
3   "scores": [
4     [
5       "其他垃圾_烟蒂",
6       "1.000"
7     ],
8   ],
9   "其他垃圾_一次性餐盒",
10  "0.000"
11 ],
12  "厨余垃圾_水果果皮",
13  "0.000"
14 ],
15  "厨余垃圾_蛋壳",
16  "0.000"
17 ],
18  "可回收物_塑料玩具",
19  "0.000"
20 ],
21 }
22
23
24
25 }
```

图 56

已经将两个模型下载到本地，见附件（best.pt 和 saved_model.pb）

2. 开发环境实践

成功运行 NoteBook

名称	状态	图像	规格	描述	创建时间	创建者	操作
notebook-538a aedddc8e-f7e5-4e95-8d80-3b216e3d4926	运行中 (57分钟)	pytorch1.8-cuda10.2...	CPU: 2核 8GB	notebook实验	2024/07/09 10:59:47 ...	hid_0cd7dsbde2ftzz	打开 启动 停止 更多

图 57

3. ModelArts Standard 自定义算法

成功训练出模型，见附件 mnist_cnn.pt

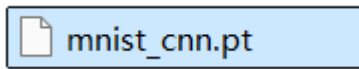


图 58

成功部署模型为在线服务，并成功预测



图 59

（三）手写数字识别

运行训练代码并检测

```
ansforms is deprecated from versio
{'Accuracy': 0.9681490384615384}
PS E:\h>
```

图 60

生成模型文件并保存到本地（见附件）

 checkpoint_lenet-1_1875.ckpt	2024/7/27 12:10	CKPT 文件	483 KB
 checkpoint_lenet-graph.meta	2024/7/27 12:10	META 文件	22 KB

图 61

四、实验存在的问题

（一）环境配置

①换盘时，在运行所有的 export、import 命令时，路径中不能含有空格，如 Program files，否则命令无法执行，所以需要找到一个路径中无空格的位置放置移动的 Ubuntu 系统。

②换盘后修改用户最开始使用了网上部分博客的指令 ubuntu2204 config --default-user your_user_name，但是默认用户并没有改变，后通过查阅其它博客，修改注册表成功。

<https://www.jianshu.com/p/636f2c47792e>（简书）

（二）端边云协同

ModelArts Standard 自定义算法

训练过程中报错

指定文件或目录不存在或无法找到 | 推荐解决方案：1、检查文件路径是否正确；

2、检查分隔符是否正确；3、检查文件权限是否正确；4、检查文件是否存在

```
381 Successfully uninstalled moxing-framework-1.17.3
382 ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following depend
383 conflicts:
384 requests 2.26.0 requires urllib3<1.27,>=1.21.1, but you have urllib3 2.0.7 which is incompatible.
385 modelarts 1.1.6 requires urllib3==1.21.1, but you have urllib3 2.0.7 which is incompatible.
386 Successfully installed lazy-import-0.2.2 moxing-framework-2.1.6.879ab0f4 urllib3-2.0.7
387 WARNING: You are using pip version 21.0.1; however, version 24.0 is available.
388 You should consider upgrading via the '/home/ma-user/anaconda3/envs/PyTorch-1.8/bin/python -m pip install --upgrade pip' command.
389 no found requirements.txt, skip requirements installing.
390 time="2024-07-26T19:28:52+08:00" level=info msg="training_pid: 333" Component=ShellScripts Platform=ModelArts-Service
391 Processing...
392 Traceback (most recent call last):
393   File "/home/ma-user/modelarts/user-job-dir/mnist-code/train.py", line 318, in <module>
394     main()
395   File "/home/ma-user/modelarts/user-job-dir/mnist-code/train.py", line 278, in main
396     convert_raw_mnist_dataset_to_pytorch_mnist_dataset(args.data_url)
397   File "/home/ma-user/modelarts/user-job-dir/mnist-code/train.py", line 206, in convert_raw_mnist_dataset_to_pytorch_mnist_dataset
398     extract_archive(os.path.join(data_url, f), raw_folder)
399   File "/home/ma-user/modelarts/user-job-dir/mnist-code/train.py", line 155, in extract_archive
400     with open(to_path, "wb") as out_f, gzip.GzipFile(from_path) as zip_f:
401   File "/home/ma-user/anaconda3/lib/python3.7/gzip.py", line 168, in __init__
402     self._fileobj = self._myfileobj = builtins.open(filename, mode or "rb")
403 FileNotFoundError: [Errno 2] No such file or directory: '/home/ma-user/modelarts/inputs/data_url_0/train-images-idx3-ubyte.gz'
404 time="2024-07-26T19:28:53+08:00" level=info msg="the run_train script is exited with 1" Component=ShellScripts Platform=ModelArts-Service
405 time="2024-07-26T19:28:53+08:00" level=info msg="command is exit with 1" file="process.go:178" Command=bootstrap/run Component=ma-training-toolkit
406 time="2024-07-26T19:28:53+08:00" level=info msg="detected pattern: map[]\n" file="process.go:197" Command=bootstrap/run Component=ma-training-toolkit
407 time="2024-07-26T19:28:53+08:00" level=info msg="analyzer is diagnosing for http request" file="analyzer.go:457" Command=analyze Component=ma-training-toolkit
408 time="2024-07-26T19:28:53+08:00" level=info msg="time required for conclude: 2 ms" file="analyzer.go:410" Command=analyze Component=ma-training-toolkit
409 2024/07/26 19:28:53 http: superfluous response.WriteHeader call from training-toolkit/pkg/diagnose.(*Analyzer).followerServeMux.func1 (analyzer.go:467)
410 time="2024-07-26T19:28:53+08:00" level=info msg="device log export result is false" file="listener.go:189" Command=bootstrap/run Component=ma-training-toolkit
411 time="2024-07-26T19:28:53+08:00" level=info msg="current job is not env job, skip parse record log" file="record_log_handler.go:318" Command=bootstrap/run
```

图 62

正确上传训练数据后解决

<input type="checkbox"/>	名称/ID	作业类型	状态	运行时长 (hh...	创建时间
<input type="checkbox"/>	job-465f-copy-d1f0 faec8870-1287-4777-933d-3976297...	训练作业	✅ 已完成	00:03:12	2024/07/26 20:...
<input type="checkbox"/>	job-465f 26bd44e8-6704-4b29-a42d-61b712...	训练作业	❌ 运行失败	00:00:43	2024/07/26 19:...

图 63

(三) 手写数字识别

安装 mindspore 时要确认 python 版本是否匹配

```
Could not solve for environment specs
The following packages are incompatible
└─ mindspore-cpu 2.0.0** is installable with the potential options
   └─ mindspore-cpu [2.0.0|2.0.0a0|2.0.0rc1] would require
      └─ python >=3.7,<3.8.0a0 , which can be installed;
      └─ mindspore-cpu [2.0.0|2.0.0a0|2.0.0rc1] would require
         └─ python >=3.8,<3.9.0a0 , which can be installed;
         └─ mindspore-cpu [2.0.0|2.0.0a0|2.0.0rc1] would require
            └─ python >=3.9,<3.10.0a0 , which can be installed;
pin-1 is not installable because it requires
└─ python 3.10.* , which conflicts with any installable versions previously reported.
```

图 64

如果在 Linux 系统中安装，则安装时不会提醒环境是否匹配，运行时才会报错

```
Traceback (most recent call last):
  File "/home/xumian/down.py", line 3, in <module>
    from mindspore import context
  File "/home/xumian/anaconda3/envs/zn/lib/python3.7/site-packages/mindspore/__init__.py", line 18, in <module>
    from . import common, train
  File "/home/xumian/anaconda3/envs/zn/lib/python3.7/site-packages/mindspore/common/__init__.py", line 16, in <module>
    from . import dtype
  File "/home/xumian/anaconda3/envs/zn/lib/python3.7/site-packages/mindspore/common/dtype.py", line 20, in <module>
    from .._c_expression import typing, EnvInstance_
ImportError: libssl.so.1.1: cannot open shared object file: No such file or directory
```

图 65

五、心得体会及建议

1. WSL2 与 Anaconda 环境的配置

通过在 WSL2 系统(Ubuntu)上安装并配置 Anaconda 环境,我掌握了在 Linux 环境下进行 Python 开发的基本技能。Anaconda 的强大包管理和环境管理功能,为我的实验提供了极大的便利。学习如何在 WSL2 中高效使用命令行工具,增强了我的系统操作能力。

2. 端边云协同的学习与实践

端边云协同作为当前技术发展的重要趋势,我通过学习这一概念,理解了其在数据处理和智能应用中的重要性。特别是在资源分配、延迟降低和系统扩展性方面,端边云协同展现出了明显的优势。

在华为云的 ModelArts 平台上,我创建了口罩检测的自动学习项目和自定义算法的手写数字识别训练作业。通过 ModelArts,我学会了如何利用云端强大的计算资源进行模型训练和部署。ModelArts 的易用性和功能丰富性,让我对云平台在机器学习领域的应用有了更深的认识。

在 ModelArts 上管理 Notebook 实例的过程中,我学习了如何高效地进行代码编写、实验记录和结果分析。Notebook 的交互式编程环境,极大地提高了我的开发效率和学习兴趣。

3. MindSpore 框架的本地训练

利用 MindSpore 框架在本地训练手写数字识别模型,是我对本地计算能力的一次深入实践。MindSpore 的高性能和易用性,让我对深度学习框架有了更全面的认识。同时,我也学习了如何针对不同的硬件平台优化模型训练过程。

4. 总结与展望

通过本次实验,我不仅学习了端边云协同的理论知识,还通过实际操作加深了对云计算、边缘计算和本地计算协同工作模式的理解。未来,我希望能够将这些知识应用到更多的项目中,不断探索和创新,为智能技术的发展贡献自己的力量。

5. 建议

课程时间相对较紧张,可以提前公布课程相关课件。华为云社区文档部分有

误，没有修改。