当MMEdu遇上行空板——“智能稻草人”项目的后续研究

谢作如　浙江省温州中学
程龙恺　上海人工智能实验室

摘要：从确定问题、收集数据到选择AI模型，最后训练出一个可行的AI模型，是中小学AI科创活动的重要流程。但在AI模型训练出来后，还需要部署在各种开源硬件上使其成为一个真正的“AI产品”或者AI应用，这才是AI科创活动的最终目标。本文以“智能稻草人”项目为例，介绍了将AI模型部署到开源硬件上的多种方案，重点介绍了搭建AI推理服务器和转换AI模型两种常见方案，并分别进行了分析和测试。

关键词：MMEdu；深度学习；AI模型部署；AI科创活动

中图分类号：G434　**文献标识码：**A　**论文编号：**1674—2117（2022）23—0077—03

问题的提出

在上一期专栏文章中，笔者设计了一个名为“智能稻草人”的科创项目。温州实验中学的老师觉得很不错，于是在全校范围开展了这个主题的项目式学习。学生们结合校园中的各种真实问题收集数据，然后借助MMEdu训练模型。学生很喜欢这个项目，研究进展顺利，模型识别效果也很不错。随着项目学习的推进，他们提出了一个新的问题：如何在开源硬件上应用MMEdu模型？

显然，这样的需求是从真实问题解决的角度提出的——因为总不能搬出一台计算机来作为运行AI模型的终端。虽然有很多品牌的开源硬件应用在中小学的创客教育中，但能够运行AI模型的开源硬件性能要求较高，需要支持Linux系统，如虚谷号、香橙派、冲锋舟、行空板等。从应用的便捷程度上来说，行空板是中小学目前最热门的选择，其自带的大屏幕很适合做各种智能作品。于是笔者将问题聚焦为如何在行空板上应用MMEdu模型。

在行空板上应用MMEdu模型的方案分析

行空板是一款拥有自主知识产权的国产教学用开源硬件，集成LCD彩屏、Wi-Fi蓝牙、多种常用传感器和丰富的拓展接口，支持常见的USB设备，接上普通的USB摄像头、USB小音箱就能完成一个智能稻草人的作品原型。如果加上舵机、电磁阀之类的执行器，则可以实现如智能灌溉、智能门禁等常见的创客作品；如果加上物联网MQTT消息的传递，则可以实现远程管理，做出智联网的作品来。

经过分析，在行空板上应用MMEdu模型有多种方案，分别介绍如下。

方案1：在行空板上安装MMEdu

所谓“解铃还须系铃人”，要想在行空板上使用MMEdu模型，最自然的想法肯定是在行空板上安装MMEdu。不过，这并非推荐的方案，因为行空板只有512M的内存，MMEdu基于OpenMMLab进行二次开发，包含了多个对系统要求较高的Python模块，环境安装比较困难。

方案2：让行空板远程调用MMEdu的推理服务

参考百度AI开放平台的做法，可以把行空板看成是一个带摄像头的Wi-Fi终端，AI模型推理工作放在一台PC机上，部署为“推理服务器”。行空板把拍摄到的照片传送给服务器，再根据返回的数据执行相应任务，如图1所示。

方案3：在行空板上部署ONNX或者NCNN环境

ONNX的全称是“Open Neural Network Exchange”，即“开放的神经网络切换”，旨在实现不同神经网络开发框架之间的互通互用。ONNX支持多平台，推理环境搭建非常方便，是部署AI应用主流选择。MMEdu支持导出ONNX模型，行空板也能够部署ONNX的推理环境。

除了ONNX外，NCNN也是可行的选择。NCNN是一个跨平台的神经网络前向计算框架，为移动端的推理做了优化，行空板已经提供了安装NCNN环境的教程，以及推理的DEMO。

● 行空板远程调用MMEdu的推理服务

1.AI推理服务器代码的实现

借助Flask或者fastapi，搭建一个类似百度AI开放平台并不困难，稍微有点Python基础的就能完成。核心代码如图2所示。

2.远程推理代码的编写

借助Request库，三四行代码就能实现将图片传送到Web服务器，并获得返回信息。参考代码如下页图3所示，其中“10.1.2.1”为Web服务器的IP地址。

作为智能稻草人作品，行空板肯定是要接上摄像头，然后定时将画面发给服务器。拍照一般使用OpenCV库，代码也非常简洁。下页图4中的代码实现了启动摄像头，拍照后传送图片到服务器，然后输出返回到信息。

3.测试情况记录

将一台笔记本电脑（CPU为I5-9750，内存为16G）设置为AI推理服务器，在启动GPU（NVIDIA GeForce GTX 1650）的情况下，对“MobileNet”的模型进行推理，平均速度是0.15秒左右，而仅仅使用CPU推理，速度也差不多。可见，AI推理在普通电脑上是没有太大的压力的，算力已经够了。

为了方便中小学的师生们部署AI应用，XEdu的开发小组特意编写了一个名为“EasyAPI”的小程序（如下页图5），只要训练好MMEdu的模型，设置必要的路径参数，即可生成Python代码，也可以直接运行，让计算机变身为AI推理服务器。

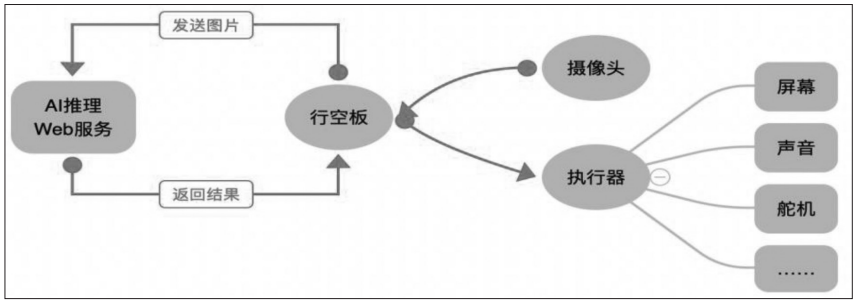

图1

```
    #导入库部分省略
app = FastAPI()

@app.post("/upload")
async def upload_file(files: UploadFile = File(...)):
    fileUpload = f"./upload/{files.filename}"
    with open(fileUpload, "wb") as buffer:
        shutil.copyfileobj(files.file, buffer)
# 网络模型的名称
    model = cls(backbone='MobileNet')
# 训练出来的模型权重文件
    checkpoint = '../XEdu/best.pth'
# 数据集的类别说明文件
    class_path = '../XEdu/classes.txt'
      model.load_checkpoint(device='cuda',checkpoint=checkpoint,class_
path=class_path)
    result = model.fast_inference(fileUpload)
    chinese_result = model.print_result(result)
    return {"tag":chinese_result[0]['预测结果'],"acc":chinese_result[0]['置信度']}

uvicorn.run(app=app,host="0.0.0.0",port=80,workers=1)
```

图2

```
    # 上传本地图片后返回信息
import requests
url = "http://10.1.2.1/upload"
files = {'files': open('qingtian3.jpg', 'rb')}
result = requests.post(url=url, files=files)
print(result.text)
```

图3

```
    # 调用摄像头后返回
import cv2
import requests
cap = cv2.VideoCapture(0)
url = "http://10.1.2.1/upload"
if(cap.isOpened()):
    ret_flag,Vshow = cap.read()#得到每帧图像
    save_url = 'save_data/test.jpg'
    cv2.imwrite(save_url,cv2.resize(Vshow,(224, 224)))
    files = {'files': open(save_url, 'rb')}
    result = requests.post(url=url, files=files)
    print(result.text)
else:
    print('摄像头未启动')
```

图4

图5

```
    Python
python ./tools/deploy.py \
    configs/mmcls/classification_onnxruntime_static.py \
    convert-config/resnet50.py \ #网络结构配置文件
     models/best_accuracy_top-1_epoch_47.pth \ #训练出的权重文件
      data/hand_draw/data/train/paper/1.png \ #任意一张输入图片
    --work-dir work_dir \ #模型转换后输出文件
    --show \
    --device cpu
```

图6

● 在行空板上部署ONNX和NCNN环境

1.环境安装

绝大多数的系统都支持ONNX，行空板上可以通过Pip命令安装“onnxruntime”。对于NCNN，行空板则提供了安装教程和编译好的文件，也可以通过Pip方式安装。

2.将MMEdu模型转换为ONNX和NCNN模型

转换模型看起来是一件很专业的事情，其实只要找到相应的工具即可。借助OpenMMLab的classification模块，只要一句命令，就能直接将模型导出为ONNX格式的模型（如图6）。

MMEdu提供了更加简单的转换方式。使用model对象的convet方法，就能直接导出指定的模型格式文件，如转换为ONNX的代码为“model.convert(backend_type='onnx')”，是不是很简单。

需要强调的是，还没有做过优化的ONNX模型在行空板上推理速度有点慢，测试结果大约是2秒，如果换成树莓派和jetsonnano，则速度应该会更快一些。这还需要进一步研究。

● 总结

“智能稻草人”项目实际上是一个范围很广的AI科创主题，几乎绝大多数的AI安防类产品原理都类似稻草人。根据解决的问题采集相应的数据，再选择合适的网络模型。在MMEdu系列工具的支持下，用AI解决问题其实并不难。对于网络结构比较复杂的模型，推荐使用方式2。从测试中可以看出，在局域网下传输图像数据，速度其实是很快的。对于模型比较简单，而且部署Web服务器比较麻烦的应用场景，则推荐使用方式3。

在完成这组AI科创案例后，笔者能够逐步明确一条中小学AI科创的学习路径，即在标准的主流的AI框架上学习训练模型，再通过相应的AI模型转换和部署工具，运行在常见的开源硬件上。这样既能够学习到主流的AI开发知识，又能很好地扩展开源硬件在AI方面的应用。当然，学习AI也不会局限于某一种开源硬件。