在行空板上部署Web前端训练的AI模型

谢作如 浙江省温州中学 程龙恺 上海人工智能实验室

摘要: 随着人工智能教育的普及,中小学的师生们不再满足于"体验",而希望通过Web页面训练出来真正有用的AI模 型,并部署到开源硬件搭建出AI应用。本文介绍了常见的Web前端模型训练平台和模型转换方法,并以行空板和ONNX为例, 展示了一个完整的A开发流程。

关键词: 深度学习, 模型转换, 模型部署

中图分类号: G434 文献标识码: A 论文编号: 1674-2117 (2023) 11-0085-03

随着人工智能的发展和普及, 深度学习已经成为人工智能的核心 内容。越来越多的教育者认识到, 如果不涉及数据采集、模型训练和 部署等内容,则几乎不能把它称为 人工智能教育。为了降低深度学习 的技术门槛,一些人工智能学习平 台设计了基于Web页面的模型训练 功能,即学生不需要编写代码,点点 鼠标就能完成从采集数据、训练模 型到推理的机器学习流程。我们将 这种模型训练方式命名为Web前端 模型训练。那么,这种基于网页前 端训练出来的AI模型,能不能像其 他模型一样,也可以部署到开源硬 件,搭建出一个真正的AI应用?针 对这个问题,笔者进行了深度探究, 并成功实现了将一个通过Web前 端训练的AI模型,转换为ONNX并 部署到行空板上。

■ Web前端训练AI模型的原 理和特点

使用Web前端方式训练模型, 需要借助TensorFlow.is这一技术。 TensorFlow.js是谷歌发布的一个 JavaScript库,它将深度学习和机 器学习技术与Web应用程序融合, 使用户能够在Web浏览器和Node. is环境中直接训练和部署机器学习 模型。简单来说, TensorFlow.js可 以让开发者用JavaScript实现机器 学习应用,通过浏览器直接与用户 交互。Teachable Machine是谷歌 创意实验室开发的小项目,可以看 成TensorFlow.js的DEMO网站。

Web前端模型训练的优点:首 先,不需要搭建专用的AI训练环 境,打开谷歌浏览器或者Edge,就 能随时随地训练模型。用户甚至可 以将Web站点搭建在本机上,不上 网都能训练。其次,这种模型训练 使用的是前端算力,即浏览器端(客 户端)电脑的算力,一个Web服务器 理论上可以支持无限用户同时训练 模型,有效解决了服务器算力匮乏 的问题。再次,借助迁移学习的方 式,仅仅提供少量图片(一个分类 20张以上),TensorFlow.js训练的 模型就能取得不错的推理效果。在 普通i5系列CPU电脑上训练,一般 在一分钟内就能完成,很适合课堂 教学。

● Web前端训练AI模型的一 般流程

在Web前端训练一个AI模型 的流程,和常见的机器学习流程是 一致的,都要经历数据采集、模型 训练和模型推理这三个环节,如下 页图1所示。

国内提供Web前端模型训练

功能的人工智能学习平台主要有 浦育、英荔和腾讯扣叮这几类。其 中,英荔平台似乎是谷歌Teachable Machine的镜像网站,除了在线训 练模型外,还能够下载训练好模 型,并且提供了TensorFlow.js模型 和h5模型(Keras框架使用的模型) 的转换。

经过分析,通过TensorFlow.js 训练的图像分类模型,采用的算法 是谷歌团队提出的MobileNet v2。 MobileNets系列的本身初衷是"for Mobile Vision Applications",是 一个轻量化的卷积网络模型,可以 显著减少计算量和参数量,同时保 持较高的准确率来提升效率,能够 运行在算力不太高的移动设备上。

■ 在行空板上部署Web前端 训练的AI模型

虽然英荔平台提供了h5格式 的模型下载,而类似树莓派、行空 板的开源硬件也能够安装Keras 的环境,似乎看起来部署模型很简 单,但实际上h5模型的加载实在太 慢了,一个简单的图像分类模型居 然要2分钟才能完成,推理速度也 很慢。笔者在《当MMEdu遇上行空 板——"智能稻草人"项目的后续研 究》一文中,已经给出了在行空板上 部署ONNX模型的方法。于是,笔 者在万能的GitHub中找到了一个 名为"tf2onnx"的库, 先将h5模型 转换为ONNX模型,然后部署在行 空板上。

tf2onnx是一个将TensorFlow



图1

```
[1] ! pip install tensorflowjs
! pip install tf2onnx
            f3dlc736498aa2f1fe3b6231ee44e76/protobuf-3.20.3-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x8
            B/s eta 0:0
          B/8 eta 0:00:00

Requirement already satisfied: typing-extensions>=3.6.2.1 in /usr/local/envs/ssenv/lib/pytho n3.9/site-packages (from onnx>=1.4.1->tf2onnx) (4.3.0)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/envs/ssenv/lib/python3.9/site-packages (from requests->tf2onnx) (2022.6.15)

Requirement already satisfied: idnac4,>=2.5 in /usr/local/envs/ssenv/lib/python3.9/site-packages (from requests->tf2onnx) (2022.6.15)
            ages (from requests->tf2onnx) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/envs/ssenv/lib/python3.9/
site-packages (from requests->tf2onnx) (1.26.11)
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/envs/ssenv/lib/python
           3.9/site-packages (from requests->tf2onnx) (2.1.1)
Installing collected packages: protobuf, onnx, tf2onnx
Attempting uninstall: protobuf
Found existing installation: protobuf 3.19.6
                      Uninstalling protobuf-3.19.6:
Successfully uninstalled protobuf-3.19.6
```

图2

```
Python
import tensorflow as tf
import tf2onnx
graph def = tf.GraphDef()
with tf.gfile.GFile('input.pb', 'rb') as f:
    graph def.ParseFromString(f.read())
with tf. Session() as sess:
    tf.import_graph_def(graph_def, name='')
    g = sess. graph
    onnx model = tf2onnx.convert.from session(sess, input names=['input:0']
output_names=['output:0'])
    with open ('output.onnx', 'wb') as f:
        f.write(onnx model.SerializeToString())
```

图3

(tf-1.x或tf-2.x)、keras、 tensorflow.js和tflite模型转换为 ONNX的工具库,可通过命令行或 Python API进行操作(如图2)。在 安装tf2onnx库的同时,还需要部 署好tensorflowjs的环境。安装过 程有点长,具体安装时间受制于网 络速度。

tf2onnx提供了多种转换方式。最方便的是调用命令行,参考命令如"python—mtf2onnx.convert—saved—model tensorflow—model—path—output model.onnx"。也可以调用Python API来完成转换,核心代码并不复杂,十来行代码即可完成。参考代码如上页图3所示。

在转换前,要先将Web前端训练的模型下载到本地。下载的模型 文件为一个压缩包,解压后可得到model.json和model.weights.bin两个文件。需要注意的是,在转换时,这两个文件要放在一个文件夹中。

● ONNX模型在行空板上的 部署测试

在完成模型转换后,就可以将这一模型部署到行空板上了。最终训练的是一个害虫的分类模型,调用行空板的摄像头进行识别。当程序启动后,在2秒内可以载入模型,每推理一帧的时间在0.3秒左右。整体项目使用BaseDeploy完成推理模块的封装,使用BaseDT完成数据预处理。一个调用摄像头的核心代码仅10行左右,参考代码如图4所示。

```
Python
import cv2
import BaseDeploy as bd
# 使用 BaseDeploy 加载一个 model
model path = './model.onnx'
mode1 = bd(mode1 path)
# 创建一个 VideoCapture 对象,参数 0 表示使用默认的摄像头
cap = cv2. VideoCapture (0)
while True:
   ret, img = cap. read()
   #使用BaseDeploy加载的模型进行推理,并设置回传为cv2格式的图像
   result, img = model.inference(img, get_img='cv2')
   cv2. imshow("frame", img)
# 释放资源并关闭所有窗口
cap. release()
cv2.destroyAllWindows()
```

图4

基于Web前端训练模型,对 当前的中小学生来说是一种很好 普及的模型训练方式。只要能够借 助低门槛的深度学习工具训练模 型,学生们用AI模型解决真实问题 的难度将大大降低。用网页训练模 型,再部署到开源硬件上,创客教 育和人工智能教育能够快速融合。

当然,TensorFlow.js也存在不足,其支持的模型类型还不是很多。但笔者相信,只要中小学生体验了用训练AI模型的方式解决了真实问题,就会自然而然对使用其他工具训练AI模型产生学习兴趣。训练AI模型本来就是流程化的工作,

每一位学生都应该了解并体验,从 而深刻理解数据、算法和算力之间 的关系。

参考文献:

[1]谢作如,程龙恺.当MMEdu遇上行空板——"智能稻草人"项目的后续研究[J].中国信息技术教育,2022(23):77-79

[2]谢作如.用新一代人工智能技术解决真实问题——谈中小学AI科创活动的开展[J].中国信息技术教育,2022(13):5-8.**②**