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AI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.js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.js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 (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。

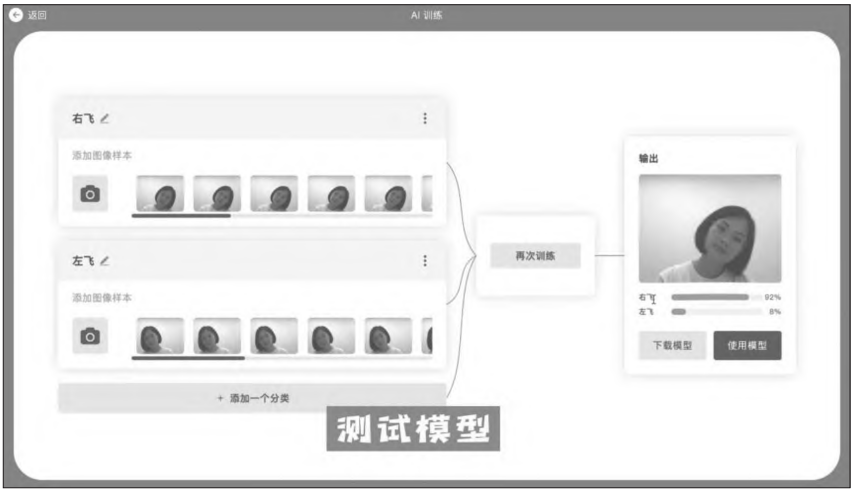

图1

```
[1] ! pip install tensorflowjs
    ! pip install tf2onnx
    f3d1c736498aa2f1fe3b623lee44e76/protobuf-3.20.3-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x8
    6_64.whl (1.0 MB)
                                                    1.0/1.0 MB 303.1 M
    B/s 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/sit
    e-packages (from requests->tf2onnx) (2022.6.15)
    Requirement already satisfied: idna<4,>=2.5 in /usr/local/envs/ssenv/lib/python3.9/site-pack
    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

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 -m tf2onnx.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'
model = bd(model_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.