

# 用PyWebIO“交互” 呈现人工智能学习成果

谢作如 浙江省温州中学

涉及学科：信息技术

人工智能是通过智能机器延伸,增强人类改造自然、治理社会能力的新兴技术,也是近几年中小学信息技术课程中最受关注的技术之一。高中信息技术课程将“人工智能初步”作为选择性必修模块。《普通高中信息技术课程标准》(2017年版2020年修订)鼓励学校建立信息技术实验室,提供开源硬件及开源AI应用框架,让学生搭建简单人工智能应用模块,并能根据实际需要配置适当的环境、参数及自然交互方式等,体验各种经典算法和应用。

绝大多数的人工智能课程,都会提供一些相关的体验活动和实验活动。最常见的是机器学习中的经典算法模型,如回归、聚类、决策树、神经网络等。借助参考代码,学生通过整理数据集,经过一定时间的训练后,一般都能顺利得到学习成果,即人工智能模型。那么,如何将这些模型部署为人工智能应用,直观形象地呈现他们的学习成果呢?

## ● 思考: 将人工智能模型部署为应用的一般方式

训练人工智能模型的最终目的是解决问题,如实现人脸识别、物体分类等。这些功能整合在一个信息系统中,就形成了智能信息系统。将人工智能模型部署为人工智能应用的前提,是具备一定的信息系统开发能力。如果仅仅将人工智能部分独立出来,部署为一个人工智能应用,实际上就是将数据输入和模型结果输出的环节剥离出来,这需要借助TK、EasyGUI或者PySimpleGUI等能够设计应用程序界面的库。如果输入的数据比较复杂,还需要QT之类的专业库支持。如果要部署为Web应用,则要采用Flask、Django之类的Web框架。

一般而言,高校学生在学习人工智能课程前都已经学过程序设计,具备一定的编程能力,如何部署AI应用不需要教师关心。但中小学不一样,由于义务教育阶段的信息科技课程标准还没有正式发布,一

些学校往往是并行开设人工智能课程和编程课程,导致学习人工智能的学生,很多并不具备相应的编程能力。即使在高中,“人工智能基础”模块是选择性必修,学生虽然已经学习过两个必修模块,但是编程技术并不扎实,用QT之类的GUI库编写复杂的程序,或者用Flask之类的Web框架编写Web应用,都有些力不从心。如果仅仅用input和Print语句进行输入输出,又很难得到直观形象的体验,相对来说,以Web形式“交互”是最好的选择。

## ● 发现: 用PyWebIO编写交互式Web应用

在本栏目的文章《用Python写一个基于Web的物联网应用程序》中,笔者曾经介绍了一个名叫“Remi”的Python库,可以在没有HTML和JS基础的情况下,快速开发出Web应用。但是,这个库的使用门槛还是稍微高了些,功能也比较复杂。笔者在比较当前众多的Python的输入输出和GUI库的基础上,最终发现了一个名叫

PyWebIO的库，非常适合应用于人工智能模型的部署环节。

1. PyWebIO库简介

PyWebIO库是一个基于Web方式来实现输入输出(I/O)的Python库。这是北京航空航天大学在读研究生王伟民同学用业余时间写的库。目前在GitHbu上获得了高达1.6K的Star。它允许用户像编写终端脚本一样来编写Web应用或基于浏览器的GUI应用，而无需具备HTML和JS的相关知识。

2. PyWebIO库的安装

PyWebIO可以采用pip命令安装，具体为：pip install remi或者pip3 install remi

3. PyWebIO库的代码示例

PyWebIO提供了一系列命令式的交互函数来在浏览器上获取用户输入和进行输出，相当于将浏览器变成了一个“富文本终端”，如图1所示。运行这段代码后，浏览器会自动打开一个本地的网址，出现如图2所示的界面。

输入姓名再点击“提交”按钮，网页上就会输出相应的文字。这种基于Web页面的“交互”，体验比黑乎乎的终端界面要好很多。

PyWebIO支持常见的网页控件。既然PyWebI的定位就是输入和输

```
from pywebio.input import *
from pywebio.output import *
# 文本输入
s = input('请输入你的名字：')
# 输出文本
put_text('欢迎你，' + s);
```

图1

出，那么也可以将网页控件分为这两类，部分控件的说明如下表所示。

尤其值得称赞的是，PyWebIO还支持MarkDown语法。除了输入输出，PyWebIO还支持布局、协程、数据可视化等特性。通过和其他库的配合，可以呈现更加酷炫的网页效果。

如果需要了解更多关于PyWebIO库的资源，请访问github或者官方文档。

Github: <https://github.com/wang0618/PyWebIO>

文档: <https://pywebio.readthedocs.io/>

● 实践：基于PyWebIO部署的人工智能作品案例

2020年，笔者和温州的其他

几位教师一起承担了浙江教育出版社的初中人工智能教材编写。首选选择了几个常见的机器学习经典案例，然后设计为实验活动，让学生体会机器学习的一般过程。一个典型的实验大致分为“数据整理”“模型搭建”“模型训练”“模型测试”四个环节，用Jupyter来完成，实验效果较好。

在学生完成实验活动后，也就是已经训练出人工智能模型后，笔者和其他几位教师又增加了“模型部署”的拓展活动环节，即利用PyWebIO来实现基于Web的数据输入和预测结果输出。学生需要增加的代码很少，效果很不错。以“手写体数字识别”实验为例，模型部署的代码如下页图3所示。



图2

PyWebIO支持的网页控件（部分）

类别	控件	代码范例
输入	文本	input(“What’s your name?”)
	下拉选择	select(‘Select’, [ ‘A’, ‘B’ ])
	多选	checkbox(“Checkbox”, options=[ ‘Check me’ ])
	单选	radio(“Radio”, options=[ ‘A’, ‘B’, ‘C’ ])
	多行文本	textarea(‘Text’, placeholder=’ Some text’)
	文件上传	file_upload(“Select a file:”)
输出	文本	put_text(“Hello world!”);
	表格	put_table([['Product', 'Price'], ['Apple', '\$5.5'], ['Banner', '\$7']],);
	图像	put_image(open(‘python-logo.png’, ‘rb’).read());
	通知消息	toast(‘Awesome PyWebIO!!’);
	文件	put_file(‘hello_word.txt’, b’ hello word!’);
	Html 代码	put_html(‘E = mc<sup>2</sup>’);

```
from pywebio.input import input,file_upload
from pywebio.output import put_text,put_image

from keras.models import load_model
import pandas as pd
import numpy as np
from keras.preprocessing import image
from keras.preprocessing.image import load_img

model = load_model('./model/3-model-vv.h5')
up_folder='./up/'

# 处理图片, 预测
def ocr(f1):
    img = load_img(f1,target_size=(28, 28),grayscale=True)
    i_img=[]
    img =image.img_to_array(img,dtype="uint8")
    i_img.append(img)
    #训练集中的图片是黑底白字, 需要进行矩阵计算, 实现图像的“反转”。
    np_image = abs(np.array(i_img)-255)
    r = np.argmax(model.predict(np_image), axis=-1)
    return r

while True:
    f = file_upload("请选择一张包含一个手写数字的图片(白底黑字):")
    f1 = up_folder + f['filename']
    webfile = open(f1,'ab')
    webfile.write(f['content'])
    webfile.close()
    put_text('你上传的文件名是: ' + f['filename']);
    put_image(open(f1, 'rb').read(),width='50px');
    # 识别并输出
    r = ocr(f1)
    put_text('AI模型的识别结果是: ' + str(r[0]));
```

图3



图4

代码中的ocr函数,本来就属于实验中给出的基本代码,实现模型根据输入的图片输出预测结果,用于“模型测试”环节。而PyWebIO实现了图片以Web上传的方式输入,模型预测结果以网页文字的形式输出,代码运行效果如图4所示。

● 反思: 从普通交互到多模态交互

Python作为人工智能领域的第一语言,虽然能够快速用于完成人工智能的各种实验,但其偏向于脚本执行的方式,内置的“交互”能力较弱,肯定需要借助于第三方的库。PyWebIO就是一个很好的例子,它能够将普通代码快速“变身”为Web应用,极大地提高了学生的学习收获感。

当然,人工智能模型的呈现也不能仅仅满足于Web的交互,还需要结合摄像头、麦克风、传感器和执行器等,实现多模态交互。这里的“模态”(Modality)指感官,如人类有视觉、听觉、触觉、味觉和嗅觉等模态。多模态交互是指人通过声音、肢体语言、信息载体(文字、图片、音频、视频)、环境等多个通道与计算机进行交流,充分模拟人与人之间的交互方式。要实现多模态交互,除了PyWebIO之外,教师还要结合OepnCV、pinpong等第三方库,这是中小学人工智能课程开发需要关注的重点之一。e