**重庆大学计算机学院**

**实习报告**

**­­­­­­­­实习单位：**

**实习岗位：**

**实习时间：**

**学 号：**

**姓 名：**

**年级专业班级：**

**目录**

实习报告参考目录，可根据具体实习进行调整

**1 实习综述**

**1.1 实习项目简介**

**1.3 实习开发技术**

**2 实习过程**

**2.1 实习时间安排**

**2.2 实习技术学习**

**2.3 实习实战过程**

**3 实习心得**

**3.1 技能与效率**

**3.2 想法与认知**

**3.3 新工具学习**

**参考文献**

格式要求：

1.报告中的一级标题二号黑体居中，二、三级标题黑体三号居左。

一级标题单倍行距、段前段后0.5行；二、三级标题行距固定值20磅，段前段后0.5行。

2.报告中正文小四号宋体，行距固定值20磅。

**1 实习综述**

**1.1实习项目简介**

随着农业产业的发展和市场需求的多样化，农产品的种类和品种日益增多，消费者对农产品的质量、安全和来源逐渐关注。然而，普通消费者往往难以准确识别和了解不同农产品的详细信息，导致选择上的困惑和不确定性。

为解决这些挑战，"农产品识别"微信小程序应运而生。该小程序使用Vision Transformer网络进行图像分类，并采用Flask框架将模型部署在后端，能够快速准确地识别各类农产品，并提供详尽的产品信息。用户只需通过手机拍摄或上传农产品的图片，即可获得关于该产品的种类、品种、产地、营养成分等重要数据，帮助消费者做出理性购买决策，同时提升农产品生产者和销售者的市场竞争力。

通过"农产品识别"微信小程序，不仅消费者能够更加便捷地获取到所需信息，生产者和销售者也能够通过数据的透明化和信息的共享，提高产品的市场认知度和竞争力。这一项目背景反映了科技创新在农业领域应用的重要性，促进了整个农产品供应链的智能化和可持续发展。

**1.2实习开发技术**

**1.2.1 Requests+Xpath数据爬取技术**

Requests 和 XPath 结合使用是一种常见的网页数据抓取和信息提取技术，特别适用于从结构化的 HTML 页面中获取特定数据。以下是简单介绍这种爬虫技术的主要步骤和关键点：

**主要步骤：**

1. **发送 HTTP 请求**：
   * 使用 Requests 库发送 HTTP 或 HTTPS 请求获取网页内容。这包括获取整个 HTML 页面或特定部分的内容。
2. **解析 HTML**：
   * 使用 XPath 选择器解析 HTML，提取出需要的数据。通常使用 lxml 库中的 etree 模块来解析 HTML，并执行 XPath 查询。

。

1. **处理和存储数据**：
   * 根据实际需求，对提取的数据进行处理、清洗或存储。可以将数据保存到文件、数据库或进行进一步的分析和处理。

**关键点和建议：**

* **XPath 表达式的编写**：
  + 确保理解目标网页的结构和节点属性，编写准确的 XPath 表达式来定位和提取所需数据。
* **异常处理**：
  + 考虑网络请求可能遇到的超时、连接中断等异常情况，编写适当的异常处理代码来增强爬虫的稳定性和健壮性。
* **合规性和道德考量**：
  + 遵守目标网站的 robots.txt 规则和使用条款，避免对网站造成不必要的负担或侵犯隐私。
* **性能优化**：
  + 考虑到爬取大量数据时的性能问题，可以采用并发请求、请求延迟控制等技术手段来优化爬虫的效率和性能。

**应用场景：**

* **数据采集和分析**：从多个网页中采集数据，用于市场调研、竞争分析等业务需求。
* **内容聚合和监控**：定期从新闻网站或博客中提取更新内容，用于内容聚合或舆情监控。
* **价格比较和产品信息抓取**：电商网站数据的抓取，用于商品价格比较或产品信息的收集。

结合 Requests 和 XPath 的爬虫技术，能够高效地从网页中抓取所需的数据，为数据驱动的应用提供了重要的支持和基础。

**1.2.2 Pytorch深度学习框架**

PyTorch 是一个开源的深度学习框架，由 Facebook 的人工智能研究团队开发和维护，旨在提供灵活和高效的深度学习开发平台。以下是对 PyTorch 深度学习框架的简单介绍：

**主要特点和优势：**

1. **动态计算图**：
   * PyTorch 使用动态计算图，这意味着计算图是按照代码运行时动态创建的。这种方式使得模型的构建过程更加灵活和直观，特别适合处理变长序列数据或动态结构的模型。
2. **易于学习和使用**：
   * PyTorch 的 API 设计简洁明了，与 Python 的语法风格高度一致，使得用户可以更快速地上手和开发。它支持动态图模式和静态图模式（通过 TorchScript），适合从实验研究到生产部署的全面应用。
3. **丰富的工具和库**：
   * PyTorch 提供了丰富的工具和库，包括神经网络层、优化器、损失函数等，以及用于图像处理、自然语言处理等领域的扩展库（如 torchvision 和 torchtext）。
4. **支持GPU加速**：
   * PyTorch 原生支持 GPU 加速计算，通过 CUDA 和 cuDNN 库实现高效的计算，能够显著提升模型训练和推断的速度。
5. **社区和生态系统**：
   * PyTorch 拥有一个活跃的社区和强大的生态系统，提供了大量的教程、文档和示例代码，帮助用户快速解决问题和扩展功能。
6. **深度整合与研究导向**：
   * PyTorch 被广泛用于学术界和工业界的研究与开发，许多最新的深度学习模型和技术都首先在 PyTorch 中实现和发布。

**使用场景：**

* **图像处理和计算机视觉**：如图像分类、目标检测、图像生成等任务。
* **自然语言处理**：如文本分类、命名实体识别、机器翻译等任务。
* **强化学习**：如智能游戏玩家、机器人控制等领域。
* **预训练模型和迁移学习**：如使用 Transformers 架构进行自然语言处理任务的预训练和微调。

**总结：**

PyTorch 提供了一个灵活、易用且功能强大的深度学习框架，适用于从学术研究到工业应用的各种深度学习任务。它的动态计算图和直观的 API 设计使得用户能够快速迭代和实验新的模型和算法，是深度学习从业者不可或缺的工具之一。

**1.2.3 Html+Js前端框架**

HTML（HyperText Markup Language）和 JavaScript（JS）是前端开发中两个核心的技术，它们结合使用可以创建交互性强、用户体验良好的网页应用。虽然它们本身不是框架，但通常与各种前端框架结合使用，以提高开发效率和扩展性。

HTML 和 JavaScript 作为前端开发的基础技术，在与各种前端框架结合使用后，能够极大地提升开发效率和用户体验。选择合适的前端框架需要根据项目的规模、复杂度以及开发团队的技术栈来决定，每种框架都有其独特的优势和适用场景，可以根据具体需求进行选择和应用。

窗体顶端

窗体底端

**1.2.4 Flask+Nginx后端框架**

Flask 和 Nginx 是两个在 Web 开发中常用的工具，分别用于构建 Web 应用和作为反向代理服务器。以下是它们的简单介绍：

**Flask：**

**描述**：

* Flask 是一个轻量级的 Python Web 框架，由 Armin Ronacher 开发，基于 Werkzeug 工具箱和 Jinja2 模板引擎。它主要用于快速开发 Web 应用和 API。

**特点**：

* **简单易用**：Flask 设计简洁，易于学习和使用，开发者可以快速上手并快速构建原型。
* **灵活性**：Flask 提供了丰富的扩展和中间件，可以根据需要灵活地扩展功能。
* **微框架**：它并不强制使用特定的工具或库，开发者可以根据项目需求选择合适的扩展和库。

**应用场景**：

* **快速原型开发**：适合于快速开发小型到中型的 Web 应用或 API。
* **微服务**：可以作为大型系统中的微服务组件使用。
* **学习和教学**：作为入门学习 Python Web 开发的理想选择。

**Nginx：**

**描述**：

* Nginx 是一个高性能的反向代理服务器和 Web 服务器，由 Igor Sysoev 开发。它以其高效的性能、低资源消耗和稳定性而闻名，广泛用于大型互联网公司的生产环境中。

**特点**：

* **反向代理**：Nginx 可以作为反向代理服务器，接收客户端请求，并将请求转发给后端的应用服务器（如 Flask 应用）处理，然后将结果返回给客户端。
* **负载均衡**：Nginx 支持负载均衡，可以根据一定的策略将请求分发到多个后端服务器，提高系统的并发处理能力和稳定性。
* **静态文件服务**：Nginx 可以高效地处理静态文件（如 HTML、CSS、JavaScript 等），提供快速的文件访问服务，减轻动态 Web 应用服务器的负载。

**应用场景**：

* **反向代理和负载均衡**：用于将请求分发到多个后端应用服务器，提高系统的性能和可用性。
* **静态文件服务**：作为静态文件的 Web 服务器，提供快速的文件访问和下载服务。
* **高并发场景**：适用于处理大量并发请求的场景，如互联网应用的前端代理服务器。

**总结：**

Flask 和 Nginx 是在 Web 开发中发挥不同作用的两个重要工具。Flask 作为 Python 的微型 Web 框架，适合快速开发和构建灵活的 Web 应用；而 Nginx 则作为高性能的反向代理服务器和 Web 服务器，提供了负载均衡、静态文件服务等功能，能够有效提升 Web 应用的性能和稳定性。在实际开发中，可以结合使用 Flask 和 Nginx，以发挥它们各自的优势，构建高效、稳定的 Web 应用系统。

**2 实习过程**

**2.1实习时间安排**

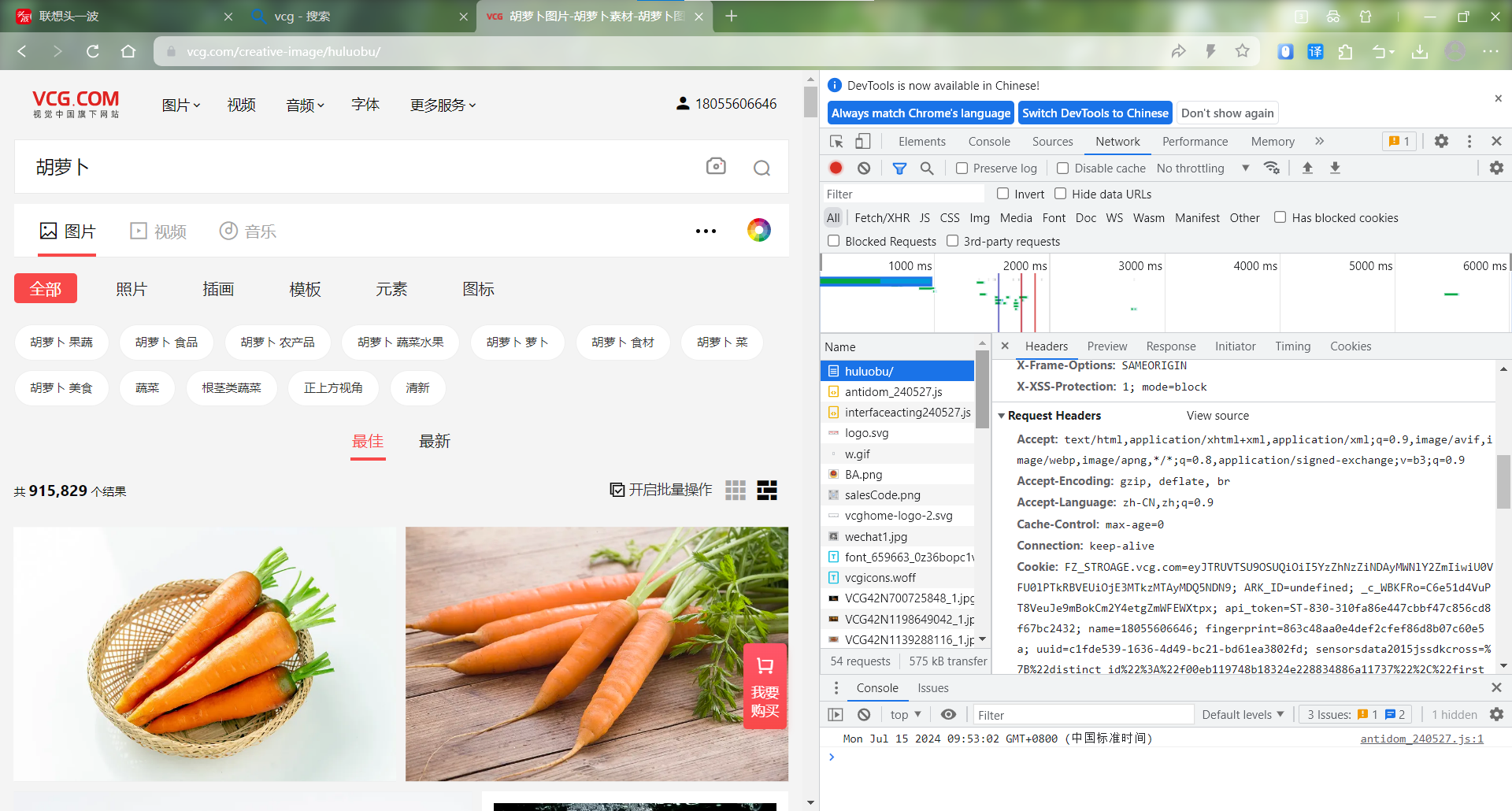
|  |  |
| --- | --- |
| 时间 | 任务 |
| 第1周 | 1. 收集图片数据集：完成 2. 对数据集进行清洗：完成 |
| 第2周 | 1. 搭建Vision Transformer网络：完成；  2. 进一步清洗数据集，并平衡不同类别的图片数量：完成；  3. 开始训练Vit模型：完成； |
| 第3周 | 1. 搭建微信小程序前端页面：完成 2. 使用Flask框架将训练好的模型部署再云服务器上：完成 3. 配置uwsgi和Nginx反向代理：完成 |
| 第4周 | 1. 解决推理程序错误率较高的问题：完成 |

**2.2实习实战过程**

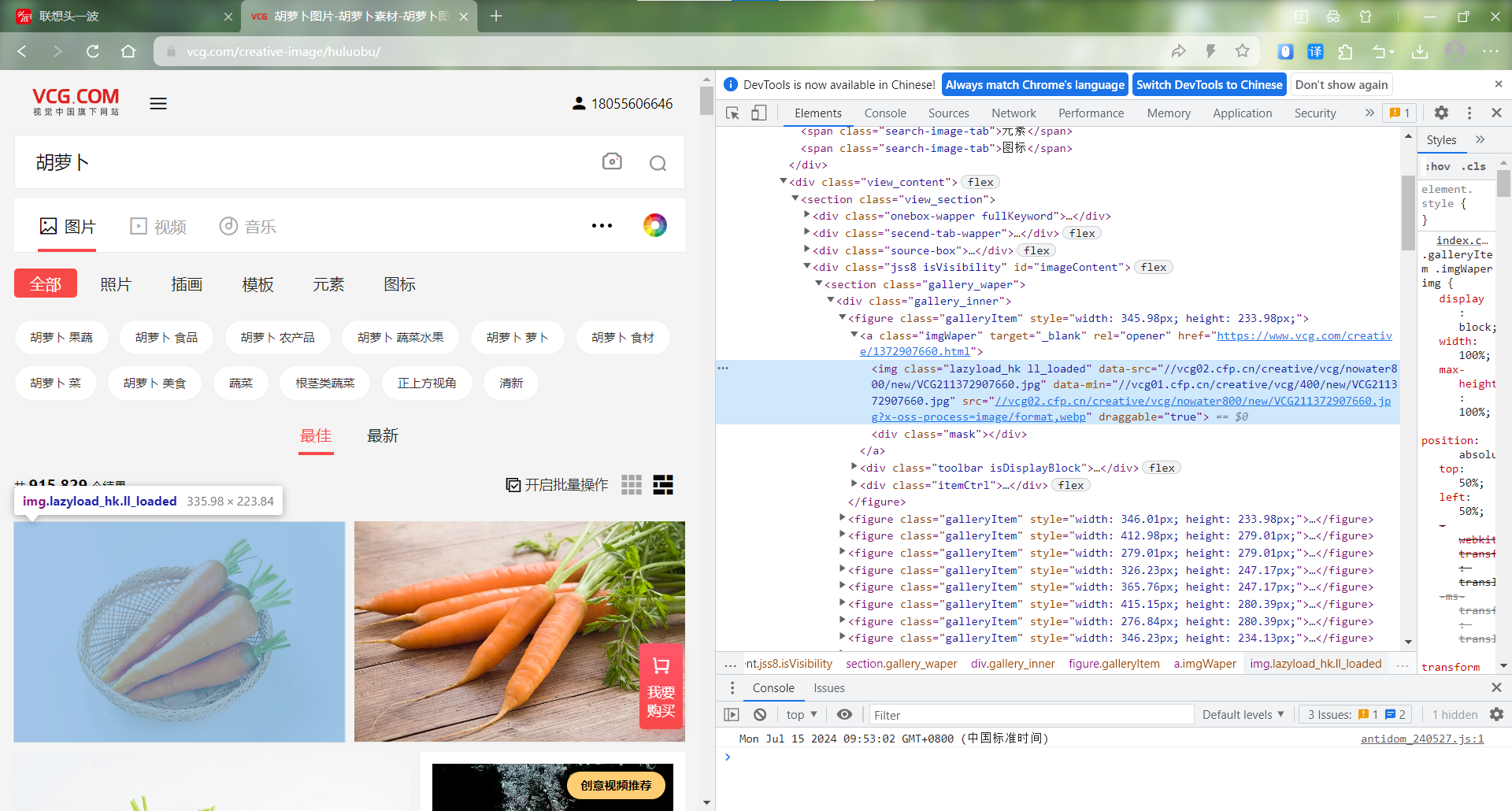
**2.2.1 数据集制作**

（1）分析目标网页

首先确定要爬取的网页，然后使用开发者工具进行抓包，查看请求头参数。

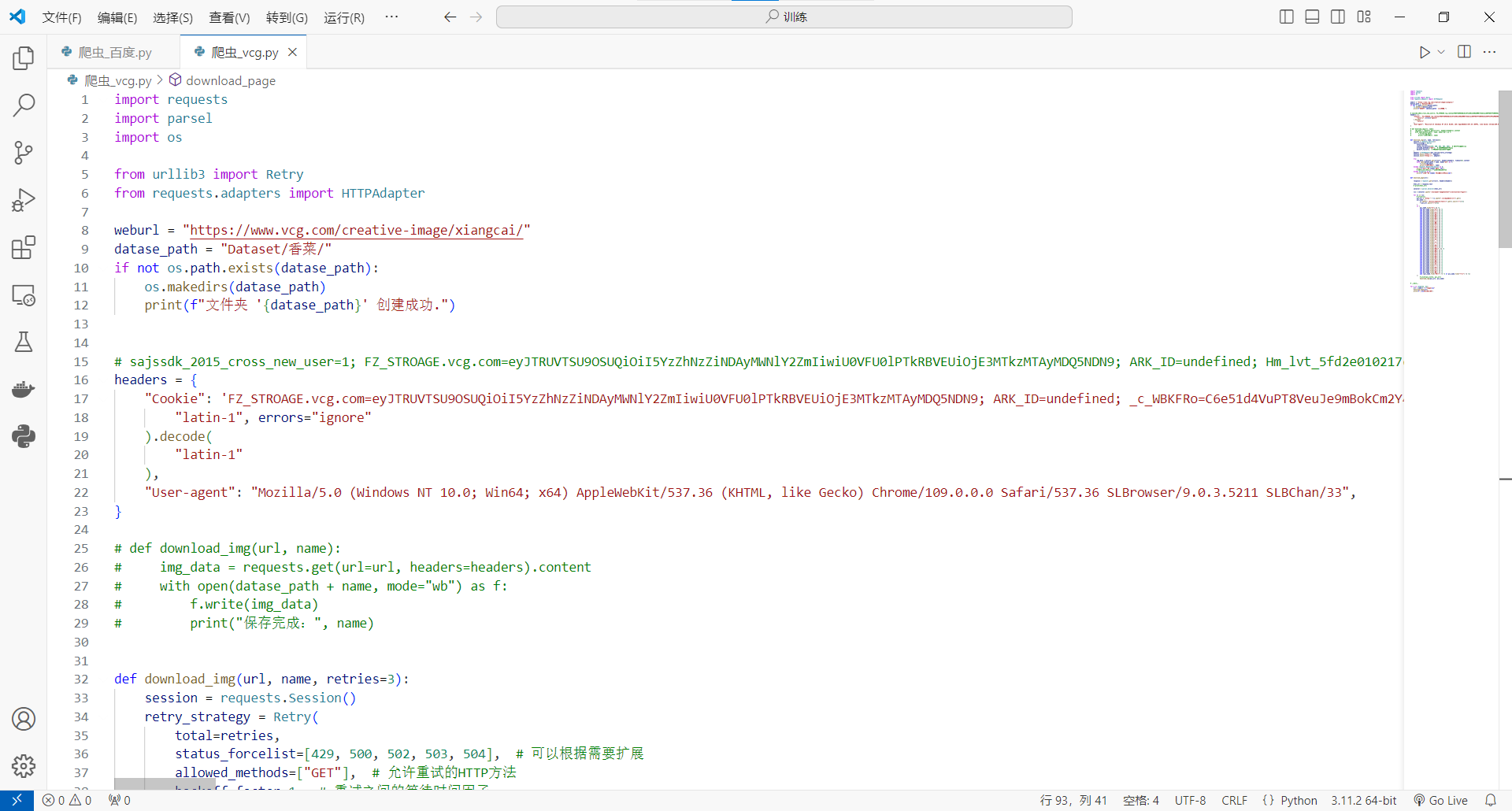


随后分析网页的html文档，找到图片资源的url

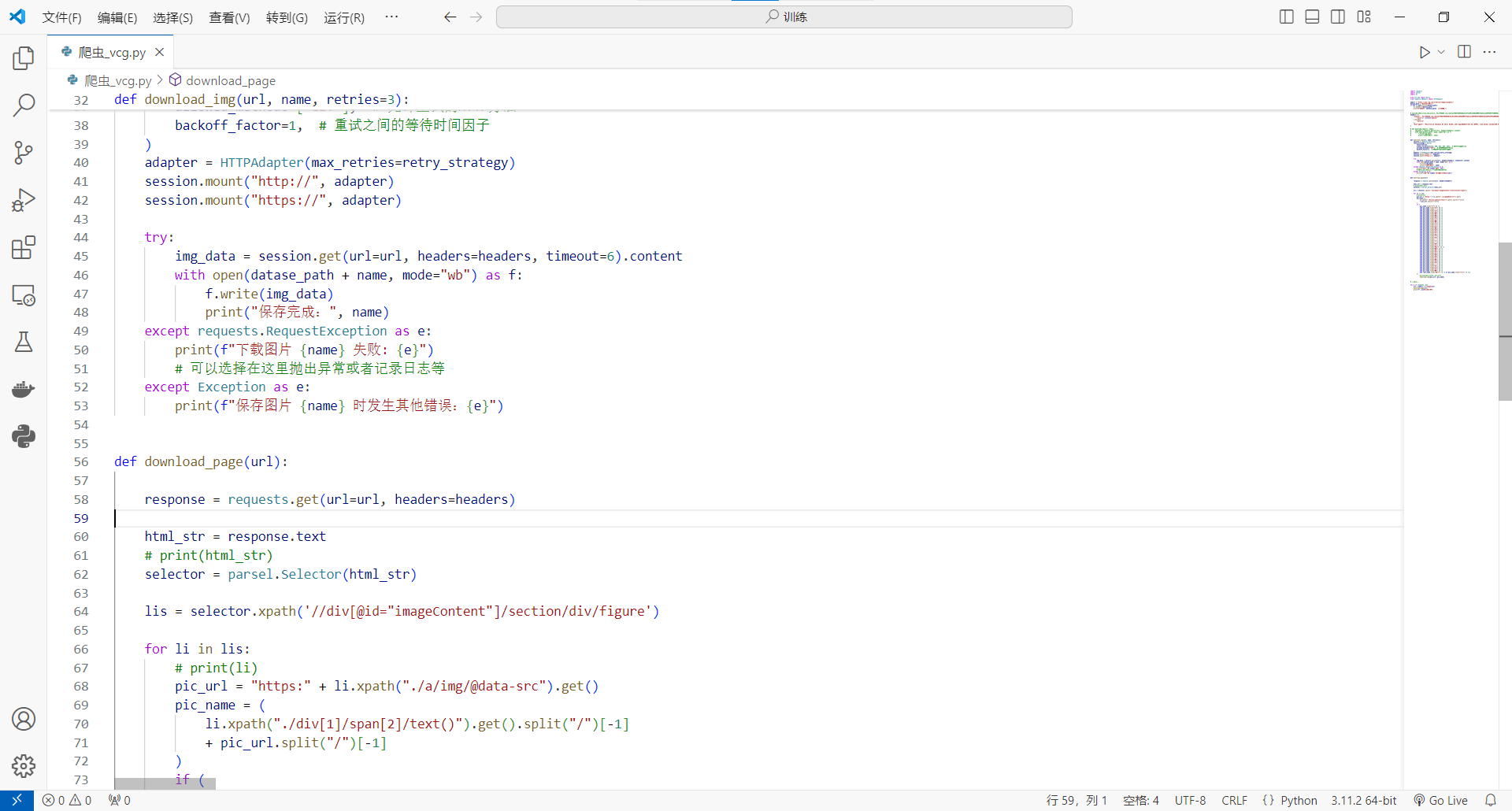


（2）针对目标网页的特征编写爬虫脚本

首先设置好请求头参数，如Cookie, User-agent等

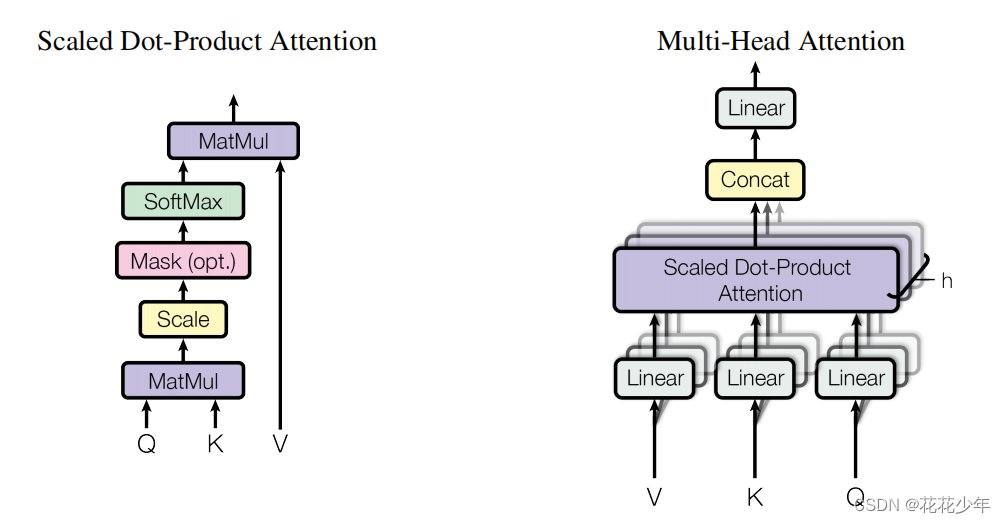


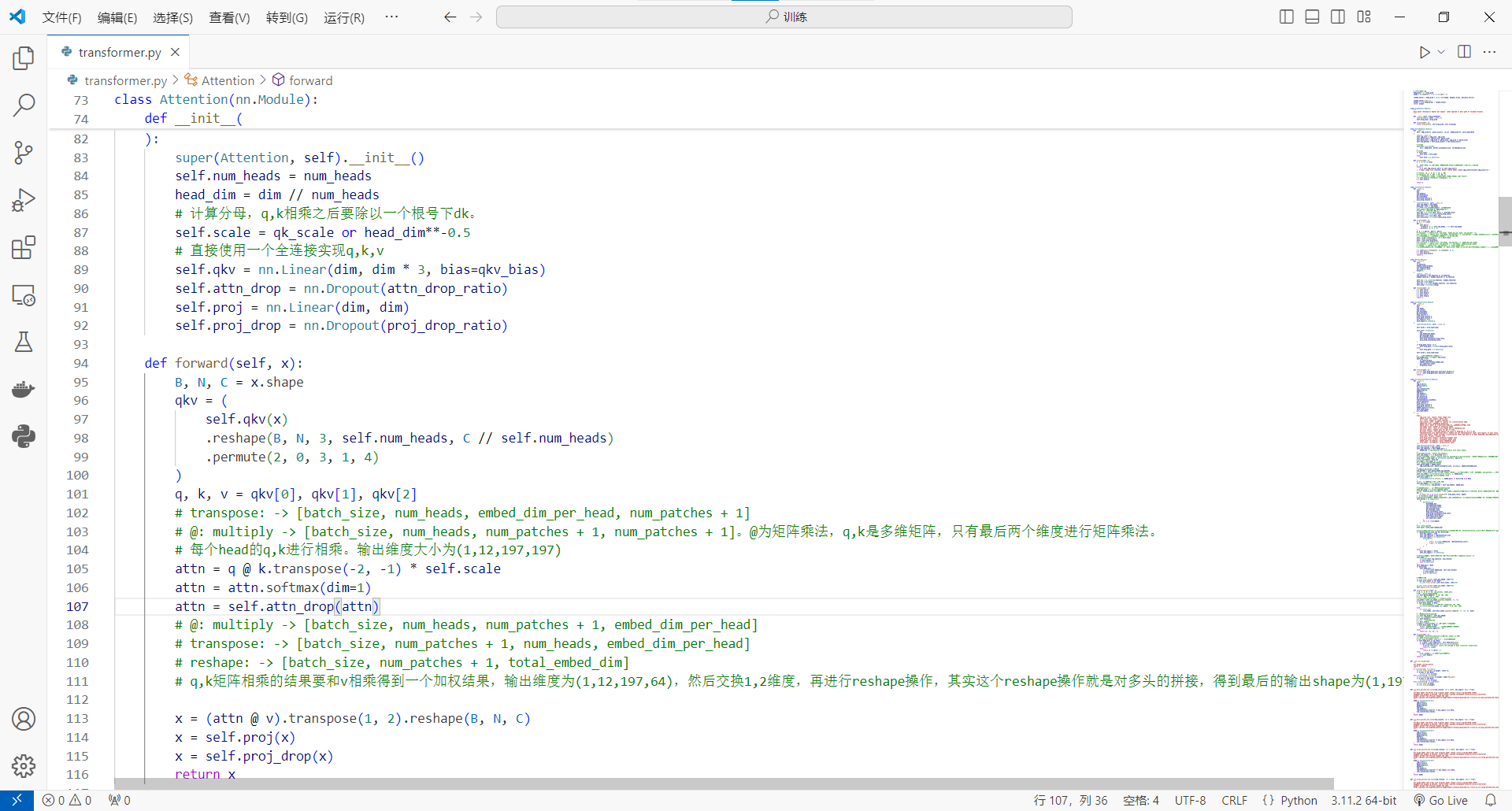
随后使用Xpath工具分析html表单，找到图片的url，下载图片并保存



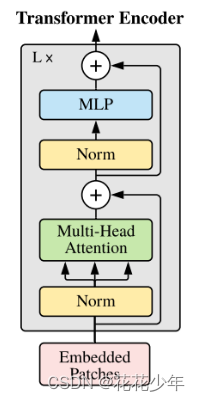
**2.2.2 搭建Vit网络**

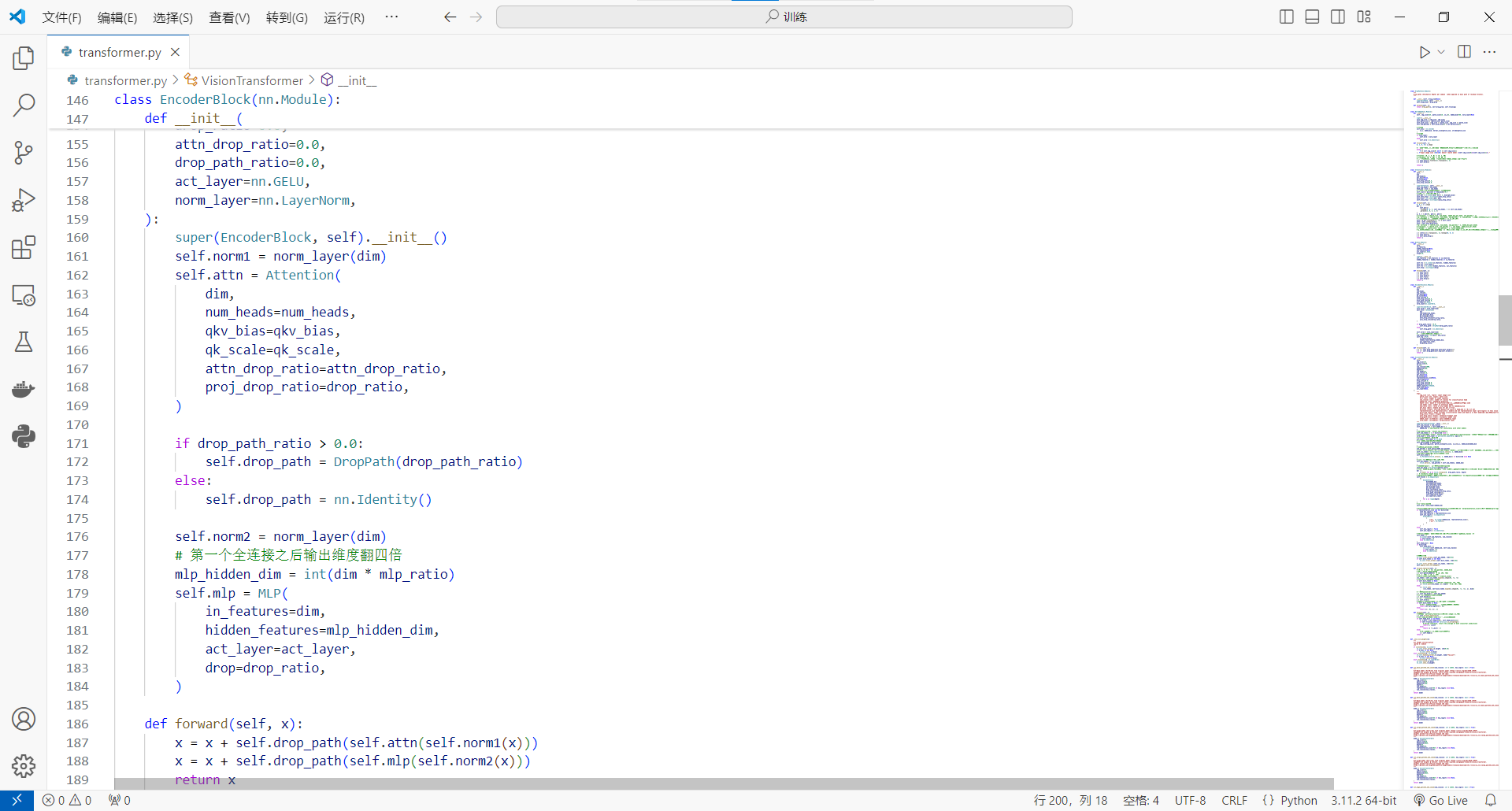
（1）实现多头注意力机制



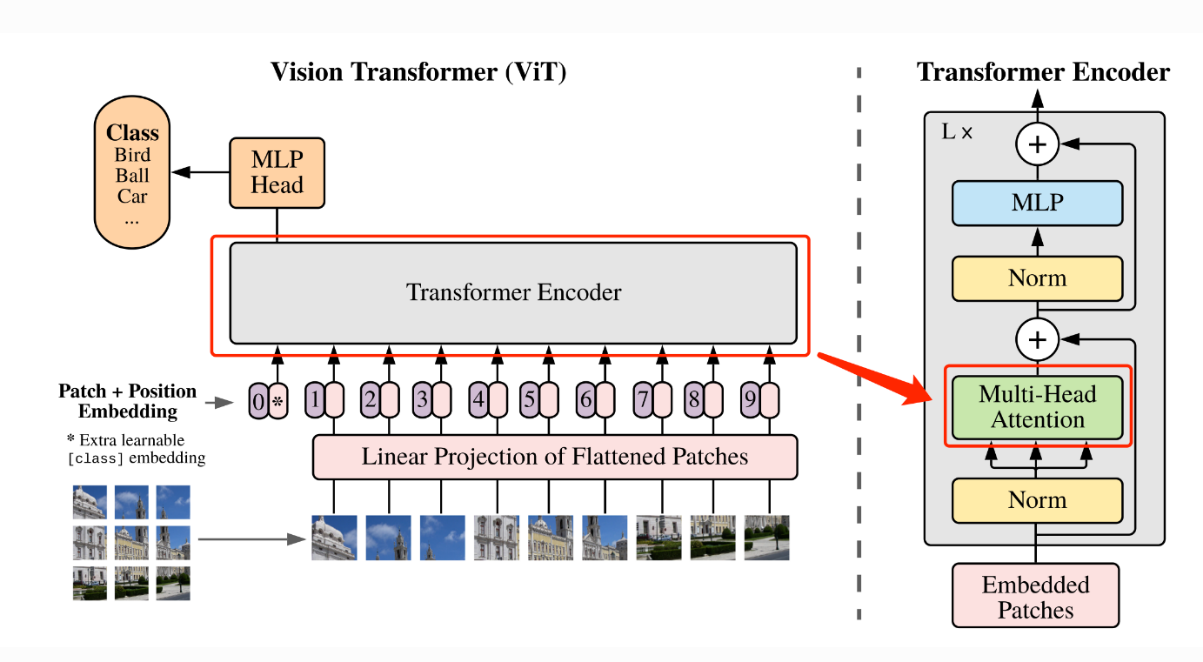


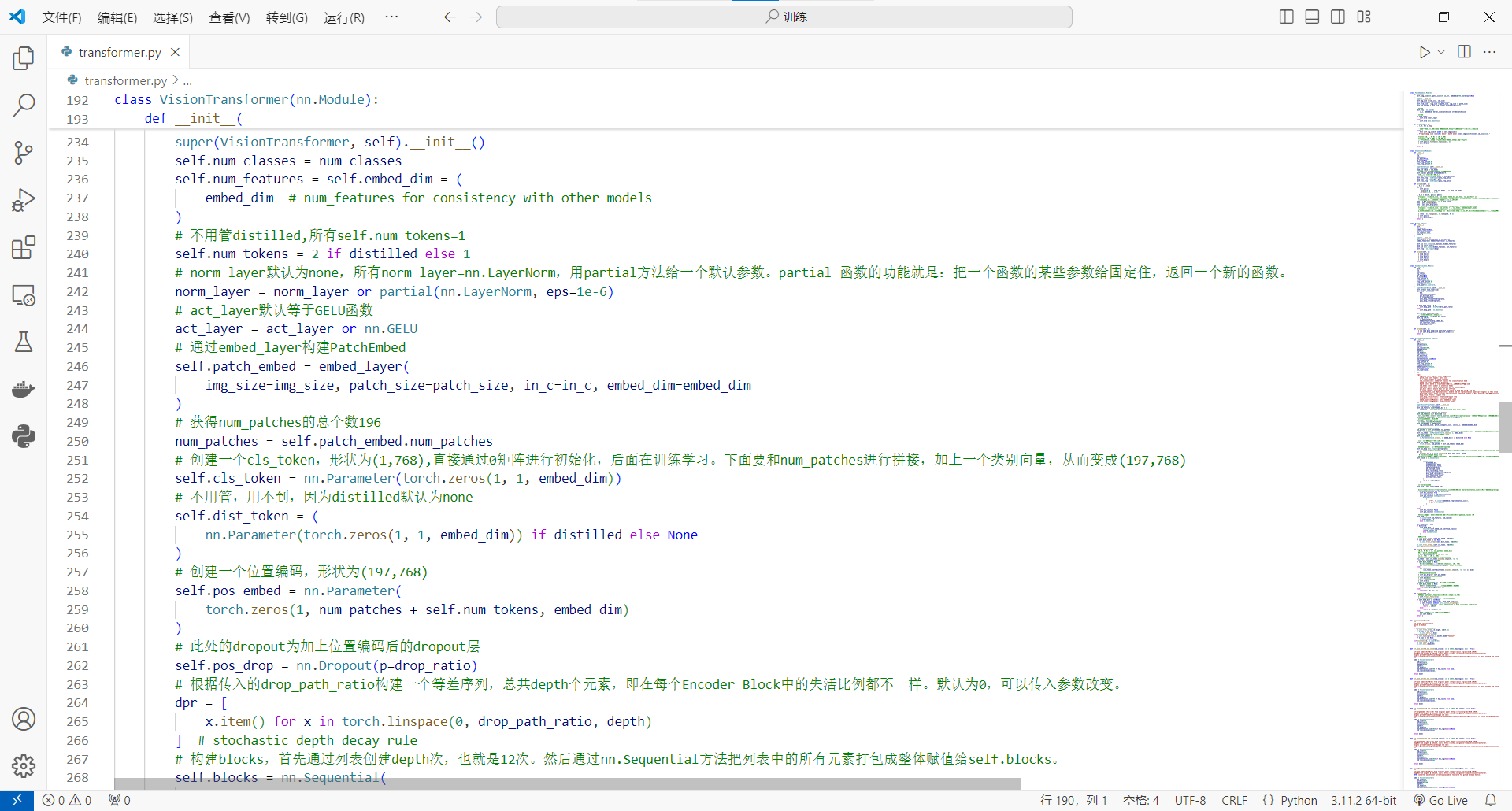
（2）实现EncoderBlock





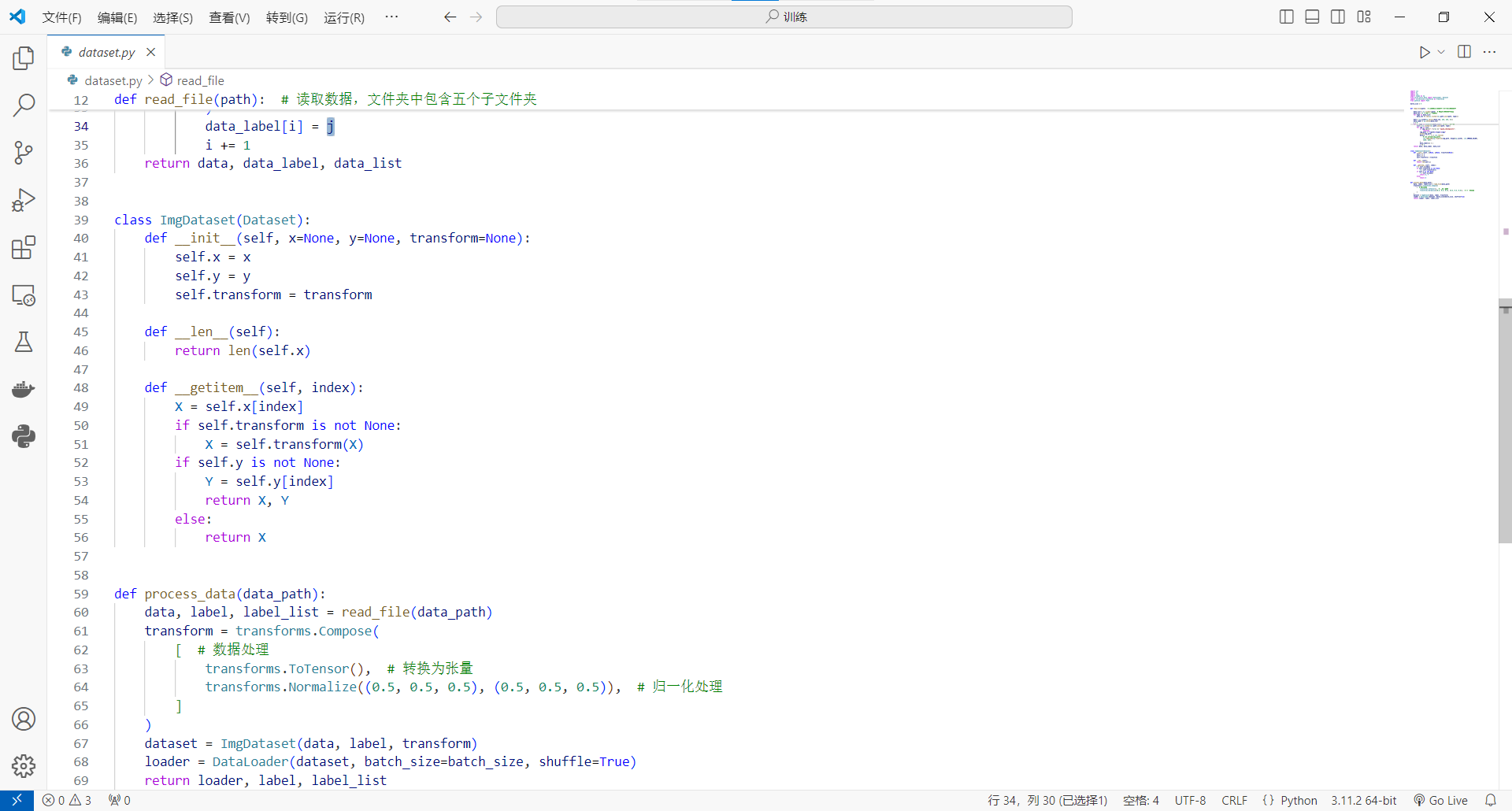
（3）搭建完整的Vit网络



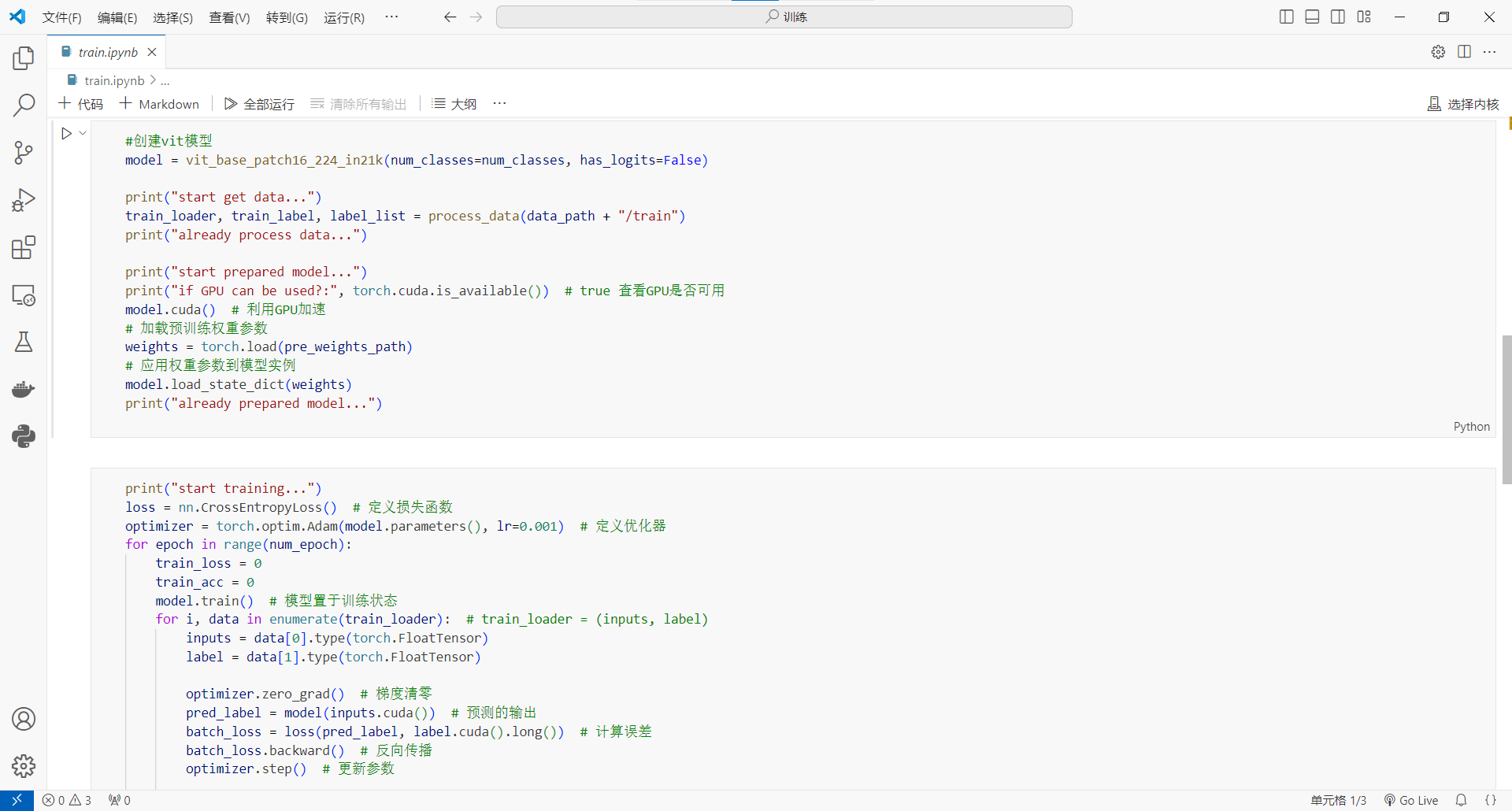


**2.2.3 开始训练**

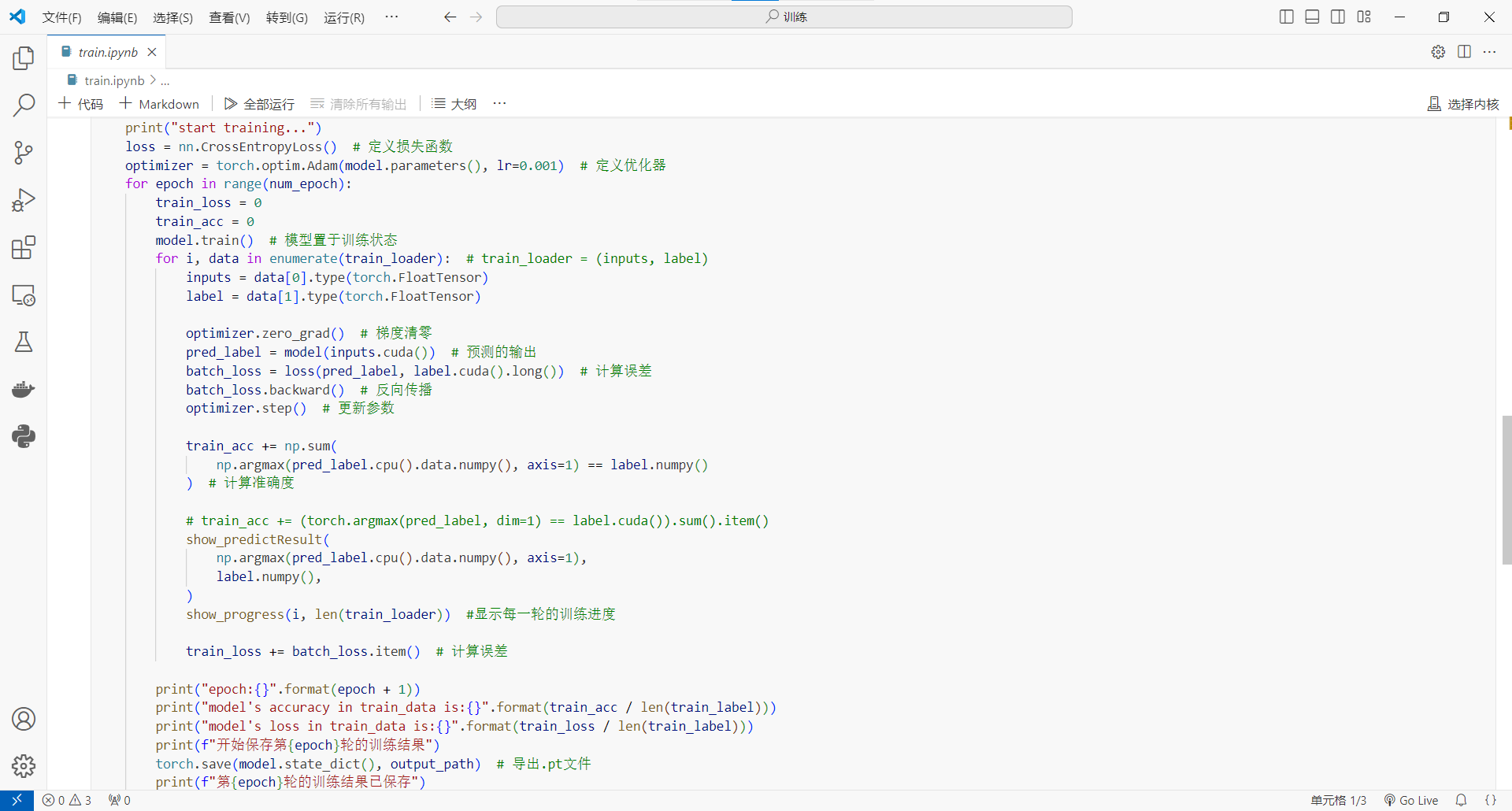
（1）预处理图片数据集



（2）加载预训练权重参数



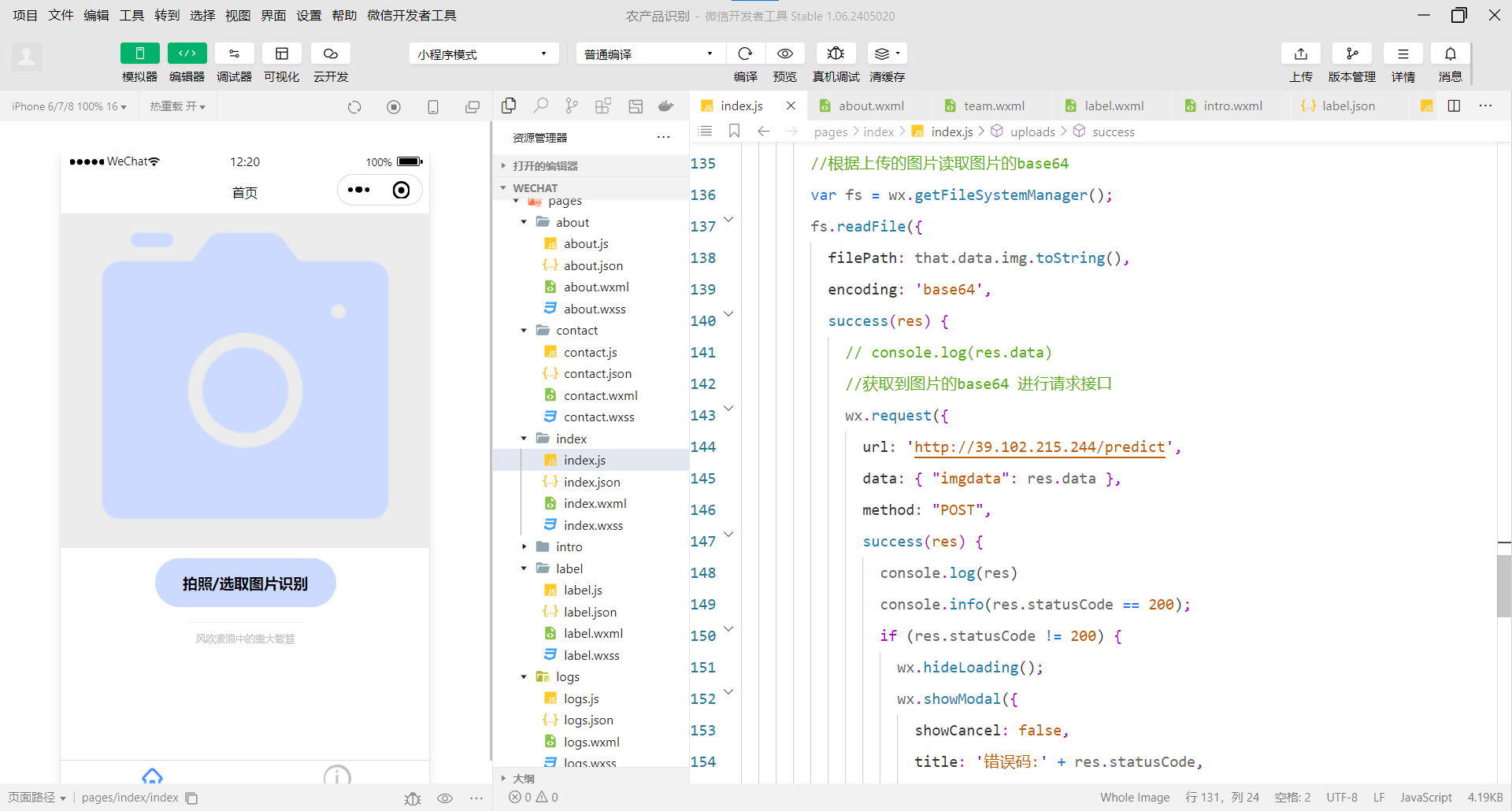
（3）开始训练并保存每轮的参数和准确率



**2.2.4 模型部署**

（1）前端开发

使用微信开发者工具编写前端页面样式和交互逻辑。将图片编码为base64格式发送给后端



（2）后端开发

使用Flask框架部署模型。将接收到的base64解码并转换为tensor张量，然后进行前向传播，将概率最高的5个类别转换为Json格式返回给前端



**2.3 成果展示**

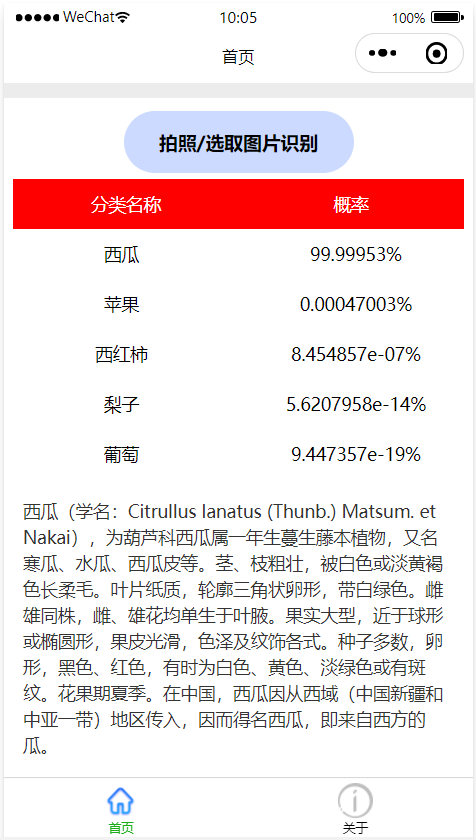
（1）minApp介绍



（2）拍照上传



（3）显示预测结果



**3.实习心得**

**3.1想法与认知**

在过去的几周里，我有幸参与了一款名为“农产品识别”的人工智能程序的开发工作。这段经历让我受益匪浅，不仅加深了我对人工智能技术的理解，还提升了我的团队合作能力和问题解决能力。

首先，我所在的团队非常注重创新和技术的前沿性。我们的项目旨在利用深度学习技术，解决实际生活中的复杂问题。在项目的早期阶段，我们花费了大量时间研究和设计模型架构，以确保其能够在不同数据集上表现出色。这期间，我学会了如何从理论到实际，将学术研究转化为可应用的解决方案。

其次，团队合作是项目成功的关键。我与来自不同背景和专业的同事紧密合作，共同攻克技术难题。通过与他们的交流和讨论，我不仅扩展了自己的视野，还学到了如何有效地与他人协作，以达成共同的目标。

在技术方面，我了解了人工智能项目的开发流程，掌握了网络爬虫、Pytorch深度学习框架，Numpy科学计算库，Flask后端应用框架和微信小程序开发等技术。尤其是在优化模型性能和处理大规模数据时，我积累了宝贵的经验，并学会了如何通过调整参数和采用新技术提升算法的效率和准确性。

最后，这段实习经历让我意识到，在人工智能领域，学习永无止境。技术不断进步，我们需要保持学习和探索的态度，不断提升自己的专业能力。我也更加坚信，人工智能技术将在未来的各个领域发挥越来越重要的作用，而我希望能在这个充满挑战和机遇的领域里，贡献自己的一份力量。

总结而言，这段实习经历不仅为我职业生涯的发展打下了坚实的基础，也让我对人工智能技术的前景充满信心。我将继续保持学习的热情和对技术创新的追求，为未来的挑战做好准备。

**3.2新工具学习**

**3.2.1 Pycharm的使用**

当谈论 PyCharm 时，我们通常指的是 JetBrains 公司开发的一款专门用于 Python 开发的集成开发环境（IDE）。PyCharm 提供了丰富的功能和工具，旨在帮助开发者提高生产力和代码质量。

**主要特点和功能：**

1. **强大的代码编辑器**：
   * PyCharm 的代码编辑器支持语法高亮、智能代码补全、代码导航、快速重构等功能。它能够识别 Python 语言的特性，并提供相关的提示和建议。
2. **调试支持**：
   * PyCharm 集成了强大的调试功能，可以方便地设置断点、查看变量的值、监控程序的执行流程，以及逐步执行代码。调试器支持 Python 的常见调试工作流程，并能与代码编辑器无缝集成。
3. **智能代码分析**：
   * PyCharm 提供了代码静态分析功能，可以检测代码中的错误、不规范的语法、潜在的性能问题等。它还能够进行代码检查并提供改进建议，帮助开发者改善代码质量。
4. **集成版本控制系统**：
   * PyCharm 支持与常见的版本控制系统集成，如 Git、Mercurial、SVN 等。开发者可以直接在 IDE 中进行版本控制操作，包括提交修改、查看历史记录、分支管理等。
5. **自动化工具和插件支持**：
   * PyCharm 提供了丰富的插件生态系统，可以根据需求安装和使用各种插件来扩展功能。它还集成了许多常用的开发工具和自动化工具，如虚拟环境管理、自动化构建工具等。
6. **跨平台支持**：
   * PyCharm 支持多种操作系统，包括 Windows、macOS 和 Linux，开发者可以在不同的平台上使用相似的开发体验。
7. **专业版和社区版**：
   * PyCharm 有两个版本，分别是专业版（PyCharm Professional）和社区版（PyCharm Community）。专业版包含了更多高级功能，如数据库工具、Web 开发支持、科学计算工具等，适合专业开发者和企业使用；社区版则是免费开源的版本，适合个人开发者和小团队使用。

**使用建议：**

* **适合的开发场景**： PyCharm 特别适合于需要进行大型 Python 项目开发的场景，比如 Web 开发、后端服务开发、桌面应用开发等。它的强大功能和集成环境能够显著提高开发效率和代码质量。
* **学习和使用成本**： 对于新手来说，PyCharm 提供了直观的用户界面和丰富的帮助文档，可以帮助快速上手和掌握基本操作。同时，由于其强大的功能和复杂的配置选项，可能需要一些时间来熟悉和掌握高级特性。

总体来说，PyCharm 是一个功能强大、灵活且高度可定制的 Python 开发工具，为开发者提供了一个全面的开发环境，支持从代码编写到调试、测试、部署的完整工作流程。

**3.2.2 Anaconda的使用**

Anaconda 是一个开源的Python和R编程语言的发行版本，用于科学计算和数据科学。它包含了许多流行的开源软件包和工具，可以简化数据分析和科学计算环境的管理和部署。

**主要特点和功能：**

1. **包含的主要组件**：
   * **Python解释器**：Anaconda 包含了 Python 解释器，是其基础和核心。
   * **Conda 包管理系统**：Anaconda 使用 Conda 作为其包管理和环境管理工具。Conda 可以安装、升级、移除包，同时管理多个环境，使得不同项目可以使用不同的 Python 版本和依赖包。
   * **科学计算包**：Anaconda 预装了许多科学计算、数据处理、机器学习相关的核心包，如 NumPy、Pandas、SciPy、Scikit-learn、Matplotlib 等。这些包通常用于数据分析、数值计算、统计建模等领域。
2. **环境管理**：
   * Anaconda 允许用户轻松地创建、管理和切换不同的 Python 环境。这对于管理不同项目的依赖关系和版本要求非常有用，避免了全局安装和版本冲突的问题。
3. **跨平台支持**：
   * Anaconda 支持多种操作系统，包括 Windows、macOS 和 Linux，使得用户可以在不同平台上保持一致的开发和执行环境。
4. **集成的开发工具**：
   * Anaconda 包含了 Jupyter Notebook 和 JupyterLab，这两个工具提供了交互式的计算环境，支持实时代码执行、可视化结果展示、Markdown 文档编写等，非常适合数据分析和教育领域使用。
5. **社区和支持**：
   * Anaconda 是一个活跃的开源项目，拥有广泛的社区支持和文档资源。用户可以通过社区论坛、GitHub 上的问题跟踪、文档等获取帮助和支持。
6. **扩展和定制**：
   * 由于 Anaconda 使用 Conda 作为包管理系统，用户可以通过 Conda 扩展包（如从 Anaconda 仓库中提供的额外包）来扩展 Anaconda 的功能。此外，用户也可以通过创建自定义 Conda 包来定制自己的 Anaconda 环境。

**使用建议：**

* **数据科学和科学计算**： Anaconda 特别适用于数据科学家、研究人员和工程师，用于数据处理、数值计算、统计分析和机器学习建模。它提供了一个完整的科学计算环境，集成了各种常用的数据科学工具和库。
* **环境隔离和依赖管理**： 使用 Anaconda 可以避免全局 Python 环境下的依赖冲突问题，使得不同项目可以独立管理其依赖关系和环境配置，提高了项目的可维护性和稳定性。
* **教育和学术研究**： Anaconda 的 Jupyter Notebook 和 JupyterLab 提供了一个交互式的学习和研究环境，非常适合教育领域的教学和学术研究，可以实时展示代码执行结果和可视化分析。

总之，Anaconda 是一个强大的工具，为 Python 和 R 用户提供了一个集成的科学计算环境，使得科学家、工程师和研究人员能够更轻松地进行数据分析和科学计算。

**3.2.3 Jupyter Notebook的使用**

Jupyter Notebook 是一个交互式的笔记本环境，支持多种编程语言，最为广泛使用的是 Python。它能够将代码、文本、图像、公式和富文本内容整合在一起，非常适合数据清洗和转换、数值模拟、统计建模、机器学习等数据科学领域的工作。

**主要特点和功能：**

1. **交互式计算**：
   * Jupyter Notebook 允许用户编写和执行代码单元（cell），每个代码单元可以独立运行并生成即时的输出。这种交互式计算方式使得用户能够逐步测试和调试代码，实时查看运行结果。
2. **支持多种编程语言**：
   * Jupyter 最初是为 Python 开发的，但现在已经支持多种编程语言，如 R、Julia、Scala 等。用户可以通过不同的 kernel（内核）来选择不同的编程语言进行编写和执行代码。
3. **丰富的展示能力**：
   * 除了代码执行结果，Jupyter Notebook 还能够显示文本、数学公式（使用 LaTeX 语法）、图像、音频、视频等多种内容。这种富文本的展示能力使得用户可以结合代码和解释性文本进行完整的数据分析和说明。
4. **易于共享和导出**：
   * 用户可以将完成的 Notebook 导出为多种格式，如 HTML、PDF、Markdown 等，便于分享和发布。此外，Jupyter Notebook 也支持将 Notebook 转换为纯 Python 脚本，方便在其他环境中执行和部署。
5. **扩展和插件**：
   * Jupyter Notebook 支持丰富的插件和扩展，用户可以通过安装各种插件来增强其功能，如语法高亮、代码补全、自动保存、主题定制等。
6. **集成其他工具**：
   * Jupyter Notebook 可以与其他数据科学工具和库无缝集成，如 NumPy、Pandas、Matplotlib、Scikit-learn 等。这使得在 Notebook 中进行数据分析、可视化和机器学习建模变得非常方便。
7. **教育和学术研究**：
   * Jupyter Notebook 在教育和学术研究领域得到广泛应用，教师和研究人员可以利用其交互性和可视化能力，展示数据分析过程和结果，提升教学和研究的效率和可视化效果。

**使用建议：**

* **学习和教学**： Jupyter Notebook 是学习和教学编程和数据科学的理想工具，可以帮助学生逐步理解代码执行的过程，并且能够结合文字说明和可视化结果进行深入解释。
* **探索和实验**： 对于数据科学家和研究人员来说，Jupyter Notebook 提供了一个探索数据、进行实验和快速原型开发的理想环境。它能够帮助用户快速迭代和验证想法，加速分析和建模过程。
* **文档和报告**： Jupyter Notebook 可以作为文档和报告的工具，将数据分析的过程、步骤和结论整理为一份完整的可交互式文档，便于团队内部和外部的沟通和理解。

总结来说，Jupyter Notebook 是一个功能强大、灵活且易于使用的工具，为数据科学家、工程师和研究人员提供了一个交互式、可视化和高效的工作环境，推动了数据科学和编程教育的发展。

**3.2.4微信开发者工具**

微信开发者工具是腾讯提供的一款专门用于微信小程序开发的集成开发环境（IDE），主要面向开发者提供了方便快捷的小程序开发、调试和发布功能。以下是关于微信开发者工具的详细介绍：

**主要特点和功能：**

1. **小程序开发环境**：
   * 微信开发者工具提供了完整的小程序开发环境，包括代码编辑器、项目管理、调试和模拟器等。开发者可以在其中编写、编辑和调试小程序的前端和后端逻辑。
2. **实时预览和调试**：
   * 开发者可以在开发者工具中实时预览小程序的效果，包括界面布局、样式和交互效果。工具提供了模拟器，可以模拟不同设备的显示效果和分辨率，便于开发者进行调试和优化。
3. **代码编辑器**：
   * 开发者工具内置了代码编辑器，支持代码高亮显示、代码补全、代码格式化等功能，提升了开发效率。同时支持 JavaScript、TypeScript、CSS 和 JSON 等常用语言的编辑。
4. **调试功能**：
   * 微信开发者工具提供了强大的调试功能，包括查看网络请求、查看日志输出、实时监测数据变化等。开发者可以在开发工具中查看小程序在真实环境中的运行情况，快速定位和修复问题。
5. **性能分析**：
   * 工具还提供了性能分析工具，可以帮助开发者分析小程序的性能指标，如页面加载时间、资源加载情况等，帮助优化小程序的性能表现。
6. **扩展能力**：
   * 开发者工具支持插件扩展，开发者可以根据自己的需求安装和使用各种插件，扩展工具的功能，如代码检查工具、模板生成器等。
7. **发布管理**：
   * 微信开发者工具还提供了小程序的发布管理功能，开发者可以在工具中完成小程序的打包、上传和发布流程。工具会自动检测代码问题和配置错误，确保小程序发布的质量和稳定性。

**使用建议：**

* **初学者**： 对于初学者来说，微信开发者工具提供了一个友好的开发环境，可以快速上手并学习小程序的开发和调试流程。
* **专业开发者**： 对于专业的小程序开发者来说，微信开发者工具提供了丰富的调试和性能分析工具，能够帮助他们更高效地开发和优化小程序。
* **团队协作**： 开发者工具支持多人协作开发，可以通过团队协作功能管理多个开发者的贡献和代码版本，提升团队的开发效率。

总结来说，微信开发者工具是一个功能全面、易用的小程序开发环境，为开发者提供了从开发到调试再到发布的全流程支持，是开发和优化微信小程序的重要工具之一。