



## 江下-惠州学院 AI 大模型特色班

### 寒假学习总结报告

完成人 : 苏 栋

班 级 : 23 大数据 1 班

学 号 : 2305070129

联 系 电 话 : 18666645846

## 1. 前言

随着人工智能技术的快速迭代，大模型已成为推动人工智能领域创新发展的核心驱动力，而 Python 编程、深度学习框架应用及版本控制工具的使用，是入门 AI 大模型领域的基础前提与核心能力。为夯实自身 AI 技术基础，系统掌握 AI 大模型相关的前置知识与实践技能，我参与了本次 AI 大模型特色班寒假培训，严格按照培训任务要求，有序推进各项学习与实践工作，确保学有所获、练有所成。

本次寒假学习以“夯实基础、强化实践、衔接进阶”为核心目标，围绕学校布置的必做任务与可选拓展任务展开，涵盖 Python 面向对象编程、PyTorch Tensor 数据处理、Git 版本控制等基础内容，同时结合自身学习兴趣，延伸学习了深度学习基础概念、卷积神经网络、实战项目开发等相关知识。学习期间，我坚持理论与实践相结合，认真研读学习资源、梳理知识体系、调试代码案例、总结学习心得，逐步提升自身的编程能力、实践操作能力与问题解决能力。

本总结报告将系统梳理本次寒假学习的具体过程、核心成果与存在不足，全面复盘学习中的思考与感悟，明确后续学习方向，为后续深入学习 AI 大模型相关知识、提升自身技术素养奠定坚实基础。

## 2. 必做任务学习情况总结

本次寒假培训必做任务聚焦 AI 大模型学习的基础能力构建，涵盖 Python 面向对象编程、PyTorch Tensor 数据处理及 Git 版本控制工具使用三大核心模块。学习过程中，我以培训提供的视频、文档资源为依托，坚持理论学习与实践操作相结合，逐点突破知识点难点，规范完成各项任务要求，切实夯实了 AI 技术入门的核心基础，具体情况如下：

### 2.1. Python 面向对象编程学习

Python 面向对象编程是 AI 开发的基础支撑，其封装、继承、多态的核心特性，能够有效提升代码的复用性、可读性与可维护性，为后续深度学习框架的应用与代码开发奠定基础。本次学习严格按照培训要求，重点研读指定视频资源的 1-16 集面向对象核心概念及 29-30 集异常处理内容，系统掌握类与对象的定义、属性与方法的封装、继承机制的实现、多态特性的应用，以及异常捕获、异常处理的核心逻辑与实现方法。

学习过程中，我重点攻克了类的继承与多态的理解难点，针对父类与子类的方法重写、抽象类的应用等问题，通过观看教学视频、编写测试代码、梳理逻辑框架的方式，逐步理清其内在关联；在异常处理学习中，深入掌握 try-except-finally 语句的使用规范，明确不同类型异常的捕获方式，有效解决了代码运行中可能出现的语法错误、逻辑错误等问题。结合学习内容，我整理了详细的学习笔记，记录知识点梳理、难点解析

及问题解决思路，并手写相关 demo 代码，通过实际编程实践巩固所学知识，确保能够熟练运用面向对象思想编写规范、高效的 Python 代码。

## 2.2. PyTorch Tensor 数据处理学习

PyTorch 作为主流的深度学习框架，其 Tensor 数据结构是实现数据处理、模型构建与训练的核心载体，掌握 Tensor 的基本操作与自动求导功能，是入门深度学习的关键环节。本次学习以李沐教学视频为核心资源，重点学习“1-1-01 课程安排”至“18-3-自动求导”的全部内容，系统掌握 Tensor 的创建、形状调整、索引与切片、数学运算等基本操作，以及自动求导的原理与实现方法，能够熟练运用 Tensor 进行数据预处理与计算。

学习过程中，我注重理论与实践的深度结合，每学习一个知识点，均通过编写 Python 脚本 demo 进行验证，重点调试 Tensor 的维度变换、广播机制、梯度计算等易出错环节。针对自动求导中梯度累积、梯度清零的逻辑的理解难点，通过查阅 PyTorch 官方文档、拆解代码运行流程、对比不同实现方式的差异，逐步掌握其核心原理；在 Tensor 数据处理实践中，遇到数据类型不匹配、维度不兼容等问题，通过调试代码、优化数据处理流程，形成了规范的问题解决思路。本次学习共整理完成多份独立的 Python 脚本 demo，详细记录了 Tensor 操作的核心代码与运行结果，并截取运行截图留存，切实提升了自身运用 PyTorch 进行数据处理的实践能力。

## 2.3. Git 使用学习与实践

Git 作为分布式版本控制工具，是 AI 开发过程中团队协作、代码管理、版本迭代的重要工具，掌握其基本操作的规范，能够有效实现学习成果的整理、归档与共享，契合培训中“将学习思考笔记与 demo 上传至指定仓库”的任务要求。本次学习以指定的 Git 学习文档为核心资源，系统学习 Git 的基础原理、核心操作，重点掌握仓库创建、文件提交、分支管理、远程仓库关联等基本操作，熟练掌握 Fork 仓库、创建个人工作目录、提交文件、发起 Pull Request (PR) 的完整流程。

实践过程中，我严格按照培训要求，首先 Fork 指定的 HZU-Jiangxia-AI-Class-2026 仓库，在个人仓库目录下，以“2305070129-苏栋”为名称创建专属工作目录，将本次寒假学习过程中整理的学习笔记、手写 demo 代码、实践项目代码等全部成果，按分类规范上传至该目录。在提交过程中，遇到文件提交冲突、远程仓库关联失败、PR 流程不熟悉等问题，通过查阅 Git 操作文档、请教同学、反复调试操作流程，逐步解决相关问题，最终完成全部学习成果的上传与 PR 提交。通过本次学习与实践，我不仅掌握了 Git 的基本操作规范，更形成了良好的代码管理习惯，为后续的学习与开发工作奠定了基础。

### 3. 可选任务学习情况总结

本次寒假培训可选任务聚焦深度学习基础概念的理解与实践应用，旨在衔接必做任务的基础能力，进一步拓展 AI 技术视野，提升深度学习相关的理论认知与实践能力。结合自身 AI 学习基础与兴趣，我主动推进可选任务的学习，在完成核心概念理解的基础上，延伸开展多项实战项目开发，既夯实了深度学习理论基础，也提升了工程实践能力，具体学习情况如下：

#### 3.1. 深度学习基础概念学习

深度学习基础概念是构建 AI 大模型认知的核心，本次学习围绕培训要求的算子、激活函数、损失函数、训练方法及优化技术五大模块展开，结合李沐、李宏毅教学视频及 d2l 书本资源，系统梳理各核心概念的定义、原理、适用场景及数学逻辑，重点突破概念间的内在关联，实现从理论认知到逻辑理解的转化。

在算子学习中，重点掌握 linear 线性层、conv 卷积层、rnn 循环神经网络、lstm 长短期记忆网络的核心原理，明确各算子的输入输出逻辑、参数设置规范，理解卷积层的特征提取机制、LSTM 对序列数据的处理优势，为后续神经网络构建奠定基础；在激活函数学习中，深入研究 sigmoid、relu、gelu、swiglu 等常用函数的数学表达式、梯度特性，对比不同激活函数在解决梯度消失、梯度爆炸问题上的差异，明确 relu 函数在深层网络中的应用优势；在损失函数学习中，系统掌握交叉熵、KL 散度、MSELoss 均方误差、L1Loss 等的计算逻辑，理解各类损失函数在分类、回归任务中的适用场景，能够根据任务需求选择合适的损失函数。

同时，我注重夯实相关数学基础，梳理各概念涉及的矩阵运算、梯度计算、概率统计等核心数学逻辑，通过推导数学公式、结合代码实践的方式，深化对理论知识的理解，避免单纯的概念记忆，切实实现理论与数学逻辑的深度融合。

#### 3.2. PyTorch 框架实践拓展

在掌握深度学习基础概念的基础上，我以 PyTorch 框架为核心，开展实践拓展学习，严格按照培训要求，实现基础概念的代码落地，同时结合自身兴趣，延伸开展多项实战项目，提升框架应用的熟练度与灵活性。

框架应用实践中，我利用 PyTorch 封装的 API，分别实现了 linear、conv 等算子的构建，完成 sigmoid、relu 等激活函数的调用与自定义实现，搭建了包含损失函数、梯度下降优化的简单神经网络，熟练掌握模型训练的基本流程，理解梯度下降算法优化网络权重的核心逻辑。针对归一化、残差链接、Dropout 等优化技术，通过编写代码验证其对模型性能的提升效果，明确各优化技术的应用场景与参数设置方法，逐步掌握 PyTorch 框架在深度学习中的核心应用能力。

结合框架学习，我开展了多项实战项目开发，核心包括手写数字识别、卷积神经网络（CNN）系列学习、停车场车位识别、银行卡号识别等项目。在手写数字识别项目中，搭建基于 PyTorch 的神经网络模型，对比不同激活函数、优化器对模型识别精度的影响，生成激活映射图与性能评估报告，深入分析模型训练过程中的过拟合、欠拟合问题及解决方法；在 CNN 学习中，系统实现 LeNet、AlexNet、VGG、GoogLeNet、ResNet 等经典网络结构，调试各网络的卷积层、池化层参数，对比不同网络的特征提取能力与训练效率，整理形成完整的 CNN 学习笔记与代码案例；在停车场车位识别、银行卡号识别等项目中，结合 OpenCV 图像处理技术与 PyTorch 模型，实现从图像预处理、特征提取到目标识别的完整流程，提升工程实践与问题解决能力。

### 3.2.1. 额外拓展学习与实践

为进一步拓展 AI 技术视野，衔接 AI 大模型相关的前沿技术，我在完成可选任务核心要求的基础上，主动开展额外拓展学习，重点聚焦生成对抗网络（GAN）与图像处理领域，完成 CycleGAN 项目的学习与实践，同时研读水下图像增强相关论文，丰富自身知识储备。

在 CycleGAN 项目实践中，我系统学习无配对图像翻译的核心原理，熟悉 CycleGAN 模型的网络结构、损失函数设计与训练流程，搭建完整的训练环境，调试模型参数，实现马到斑马、边缘到猫咪等图像风格迁移效果，同时整理项目相关的代码、文档与运行结果，深入理解生成对抗网络的核心逻辑与应用场景。在图像处理领域，利用 OpenCV 库开展图像预处理、特征提取、目标检测等实践，完成图像滤波、轮廓检测、特征匹配等操作，结合实战项目优化图像处理流程，提升图像数据预处理能力。

此外，我还研读了多篇水下图像增强相关学术论文，梳理该领域的研究现状、核心技术与发展趋势，记录论文中的核心算法与创新点，结合自身深度学习知识，思考算法的实现思路，为后续深入学习 AI 大模型与计算机视觉融合应用奠定基础。

## 4. 学习成果汇总

本次寒假 AI 大模型特色班学习，严格遵循培训任务要求，坚持理论学习与实践操作深度融合，在完成必做任务、推进可选任务的基础上，主动开展拓展学习，形成了涵盖笔记整理、代码开发、实战项目、学术研读的系统化学习成果，切实实现了基础能力的夯实与技术视野的拓展，所有成果均按规范整理归档于个人 Git 仓库（2305070129-苏栋目录下），具体汇总如下：

### 4.1. 笔记成果

围绕本次学习的核心内容，结合培训提供的视频、书籍资源及自身学习思考，整理形成了一套完整、规范的学习笔记，涵盖必做任务、可选任务及拓展学习的全部核心知识点，兼具理论性与实用性，为后续复习巩固与知识拓展奠定基础。

笔记内容具体包括：Python 面向对象编程学习笔记（涵盖类与对象、继承多态、异常处理等核心知识点）、Git 使用学习笔记（详细记录仓库操作、PR 提交等流程）、PyTorch Tensor 数据处理笔记（梳理 Tensor 操作、自动求导等核心内容）、深度学习基础笔记（包含算子、激活函数、损失函数等概念解析）、卷积神经网络学习笔记、手写数字识别项目笔记等；同时，整理了《动手学深度学习》书籍相关知识点摘要，研读并归档了 20 余篇水下图像增强相关学术论文（如《Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks》《Joint Residual Learning for Underwater Image Enhancement》等），记录论文核心算法、创新点及研究思路，形成了初步的学术研读能力。所有笔记均按知识模块分类归档，重点记录知识点梳理、难点解析及学习思考，避免流水账式记录，切实体现学习过程中的认知与感悟。

## 4.2. 代码成果

以 Python、PyTorch、OpenCV 等为核心工具，结合学习内容与实战需求，开发完成了一系列 demo 脚本与实战项目代码，所有代码均按规范编写、分类归档，具备良好的可读性、可复用性，切实将理论知识转化为实践能力，具体分为三大类：

一是必做任务相关代码，包括 Python 面向对象编程 demo、PyTorch Tensor 数据处理脚本（涵盖 Tensor 创建、维度变换、自动求导等核心操作），均已整理为独立的 Python 脚本与 Jupyter Notebook 文件，附带运行结果截图，完整复现了必做任务的学习内容；二是可选任务相关代码，包括 PyTorch 实现 linear、conv 等算子、各类激活函数与损失函数的 demo，以及手写数字识别项目相关代码（含模型构建、训练、评估等完整流程，生成了激活映射图、性能评估报告等）；三是拓展项目代码，涵盖卷积神经网络系列代码（实现 LeNet、AlexNet、VGG、GoogLeNet、ResNet 等经典网络结构）、CycleGAN 项目完整代码（含数据处理、模型搭建、训练测试等全部模块）、图像处理相关代码（基于 OpenCV 实现图像滤波、轮廓检测、特征匹配等操作），以及停车场车位识别、银行卡号识别、文档扫描 OCR 识别等实战项目代码，所有代码均经过反复调试，确保可正常运行，并按项目模块分类归档于“代码”目录下。

## 4.3. 能力提升

通过本次寒假系统学习与实践，自身在 AI 相关的理论认知、编程实践、工具使用等方面均实现了显著提升，形成了较为扎实的基础能力与初步的工程实践思维，具体体现在三个方面：

其一，理论认知层面，已系统掌握 Python 面向对象编程的核心特性、PyTorch Tensor 数据处理的核心逻辑、Git 版本控制的基本操作，深入理解深度学习的基础概念（算子、激活函数、损失函数等），初步掌握卷积神经网络、生成对抗网络的核心原理，对 AI 大模型的前置知识体系形成了清晰的认知，

其二，实践操作层面，能够熟练使用 Python 编写规范代码，熟练运用 PyTorch 框架实现数据处理、模型构建与训练，掌握 OpenCV 图像处理的基本方法，能够独立完成简单的深度学习实战项目开发，具备了基础的代码调试与问题解决能力。

其三，工具使用与学习方法层面，熟练掌握 Git 版本控制工具的核心操作，能够规范完成仓库 Fork、文件提交、PR 发起等流程，形成了良好的代码管理习惯；同时，掌握了高效学习方法，能够自主查阅官方文档、学术论文解决学习中遇到的问题，为后续深入学习 AI 大模型相关知识奠定了坚实的能力基础。

## 5. 学习过程中的问题与反思

本次寒假 AI 大模型特色班学习，是一个的渐进过程。在推进必做任务、拓展可选任务及开展额外实践的过程中，由于自身基础储备、学习方法及实践经验的局限性，不可避免地遇到了一系列知识点难点与实践操作问题。通过针对性排查、查阅资料、反复调试及总结梳理，多数问题得到有效解决，同时也结合问题背后的深层原因，进行了深刻反思，明确了自身不足与后续改进方向，具体如下：

### 5.1. 学习过程中遇到的主要问题

结合本次学习的核心模块，遇到的问题主要集中在理论理解、实践操作及学习规划三个层面，均为 AI 入门阶段常见的典型问题，具体如下：

一是深度学习理论与数学逻辑的衔接不足。在学习 PyTorch 自动求导、卷积层特征提取及损失函数计算等内容时，由于对矩阵运算、梯度下降的数学原理掌握不够扎实，导致对部分知识点的理解无法深入解释其内在逻辑。例如，在调试 Tensor 自动求导代码时，多次出现梯度计算异常、梯度清零不彻底的问题，初期仅能通过复制示例代码解决，无法自主排查问题根源；在理解卷积层的 Padding 与 Stride 参数设置时，对特征图尺寸计算逻辑的推导不够熟练，影响了对网络结构设计的理解。此外，在研读水下图像增强论文时，对论文中涉及的频率域分析、卷积神经网络改进算法等内容，因缺乏相关数学基础与前沿知识储备，难以快速把握核心创新点。

二是实践操作中的细节失误与工程思维欠缺。在代码开发与工具使用过程中，多次因细节把控不到位导致问题出现，且缺乏系统性的调试与优化思维。在 Git 操作中，初期因对远程仓库关联、分支管理及 PR 提交流程不熟悉，出现文件提交冲突、仓库关联失败等问题，甚至误提交无用文件，影响了成果归档的规范性；在 PyTorch 代码调试中，频繁出现 Tensor 数据类型不匹配、维度不兼容、模型参数设置不合理等问题，例如在搭建 CNN 模型时，因未正确设置卷积层与池化层的输出维度，导致模型训练报错，且初期排查效率较低；在 CycleGAN 项目实践中，因对训练环境配置、数据集处理流程不熟悉，出现训练过程中断、生成效果不佳等问题，缺乏对项目整体流程的把控能力。

## 5.2. 问题解决方法与实践

针对上述问题，结合学习资源与自身实际，采取了针对性的解决措施，逐步提升问题解决能力，具体如下：

针对理论与数学逻辑衔接不足的问题，调整学习节奏，加强数学基础的补充学习，结合 d2l 书本与教学视频，逐点推导自动求导、卷积运算等核心知识点的数学公式，梳理梯度计算的内在逻辑；遇到难以理解的知识点，反复观看教学视频片段，查阅 PyTorch 官方文档与相关技术博客，同时结合简单的 demo 代码，深化对知识点的理解。例如，针对 Tensor 自动求导异常的问题，拆解代码运行流程，打印每一步的梯度值，逐步排查出梯度清零不及时、数据类型错误等问题，同时整理相关调试方法，形成笔记便于后续参考；针对论文研读中的难点，先梳理论文的整体框架，重点关注实验设计与结果分析，再逐步学习核心算法，必要时查阅相关综述类论文，补充相关知识储备。

针对实践操作中的细节失误与工程思维欠缺的问题，注重规范操作流程。在 Git 操作中，先系统梳理操作流程，记录关键步骤与注意事项，遇到问题时，查阅 Git 官方文档与操作教程，逐步掌握冲突解决、PR 提交的规范方法，同时规范文件命名与归档逻辑，避免误提交、乱提交的情况；在代码开发中，注重代码的规范性与可读性，编写代码前先梳理逻辑框架，明确变量定义、函数功能，调试时采用“分段调试、逐步排查”的方法，及时记录常见报错及解决方法，形成调试笔记。例如，在解决 CNN 模型维度不兼容的问题时，逐步打印每一层的输出维度，排查出卷积层与池化层参数设置的问题，同时总结出特征图尺寸计算的规律，提升后续模型搭建的效率；在 CycleGAN 项目中，参考官方文档与开源代码，逐步调试训练参数，优化数据集处理流程，解决训练中断、生成效果不佳的问题，同时梳理项目开发的完整流程，提升工程实践能力。

针对学习规划与时间分配不合理的问题，重新梳理学习任务，制定详细的学习计划，明确各模块的学习时间与优先级，优先保证必做任务的学习质量，再逐步推进可选任务与拓展学习。合理分配理论学习与实践操作的时间，每完成一个知识点的学习，预留足够的时间进行复盘与总结，避免“赶进度”式学习；在开展多个实战项目时，明确每个项目的学习重点与时间节点，优先完成核心功能的开发与调试，再逐步优化细节，确保项目质量与学习效果。例如，调整学习计划，优先夯实 Python 面向对象编程与 PyTorch 基础，再推进 CNN、CycleGAN 等拓展项目，同时每周预留固定时间进行复盘，总结本周学习中的问题与收获，提升学习效率。

## 6. 未来学习规划与总结

本次寒假 AI 大模型特色班的系统学习，既是对 AI 入门基础能力的夯实，也是对深度学习领域的初步探索，更是自身认知与能力提升的重要契机。结合本次学习的成果、

遇到的问题及反思，为进一步弥补自身不足、衔接 AI 大模型相关领域的进阶学习，明确后续学习方向与路径，制定以下学习规划，同时对本次寒假学习进行全面复盘与总结。

在知识与能力层面，通过本次学习，我系统掌握了 Python 面向对象编程、PyTorch Tensor 数据处理、Git 版本控制等核心基础技能，深入理解了深度学习的基础概念与核心原理，初步掌握了卷积神经网络、生成对抗网络的实践方法，完成了多个实战项目的开发与调试，逐步提升了理论认知、编程实践与问题解决能力；同时，通过研读学术论文、参与项目实践，拓宽了 AI 技术视野，培养了初步的工程思维与学术研读能力，所有学习成果均按规范归档，实现了学习过程的可追溯、可复盘。在学习过程与感悟层面，本次学习让我深刻认识到，AI 领域的学习是一个循序渐进、理论与实践深度融合的过程，没有扎实的基础，就无法实现进阶提升；没有反复的实践与反思，就无法将理论知识转化为实际能力。