

培养方案

一、培养能力：

1. Python 代码能力：

- (1) 能够理解变量，常量，循环，判断
- (2) 能够理解面向对象编程，类的继承，类的属性，魔法方法
- (3) 能够理解迭代器，装饰器，列表生成器等高级用法
- (4) 能够理解使用多进程，多线程，异步等并行运算方法（非必要）
- (5) 能够通过调试，自己查找代码当中报错并解决
- (6) 能够有自学 Python 自带包的能力如 numpy, pandas

numpy: 将列表转化为矩阵进行运算从而实现加速

学习方向：如何生成多维数组（通过列表嵌套构造，通过 shape, size 初始化）

多维数组之间如何进行运算，+，-，*，/

矩阵的 shape 和 size 如何转化

后续使用 pytorch 也有大量的矩阵运算，其程序编写逻辑和 numpy 几乎类似，在 numpy 打好基础，有利于学好 pytorch，以及 matlab 等

pandas: 用于处理表格和混杂数据，更偏向于数据分析，在实际工程项目如数据分析，数据挖掘等项目，会有广泛的应用（做科研暂时不需要）

PIL: 图像处理库，常用于计算机视觉任务当中的预处理操作，以及常规图像处理操作

OpenCV: 图像处理库，常规图像处理操作

torch: 深度学习库，是科研常用的深度学习框架，后期搭建网络时，绝大多数论文对应的代码均是使用 torch 的，要求明白当中例如卷积层、批量归一化操作等方法的调用。

2. 深度学习基础理论：

- (1). 对神经元，卷积，池化，跳连接，梯度下降，反向传播，隐藏层，正则化，数据

增强, dropout, 全连接, Batch Normalization 等基础知识点理解 (基于从数学和逻辑层面的理解)

(2).对计算机视觉领域经典网络, 要对其应用场景、优缺点与核心思想等有大致了解。(可结合 blog 或 B 站讲解视频进行学习)

推荐学习:

- 经典网络: AlexNet、VGG、Res-Net
- 语义分割: UNet 系列

[UNet 家族最强系列 | UNet、UNet++、TransUNet 与 SWin-UNet 究竟哪个更强!!! -腾讯云开发者社区-腾讯云 \(tencent.com\)](#)

- 目标检测: yolo 系列 v5, v8

[物体检测算法怎么做的改进? 迪哥精讲 YOLOv1~v5、EfficientNet、TransformerDETR、YOLOv7 算法模型+改进细节+代码复现_哔哩哔哩_bilibili](#)

[吹爆! 迪哥一小时带你吃透 YOLOV8 目标检测算法, 手把手教你搞定检测、分类、追踪等视觉任务! \(人工智能/深度学习/计算机视觉\)_哔哩哔哩_bilibili](#)

- 注意力机制&Transformer

[Transformer 论文逐段精读【论文精读】_哔哩哔哩_bilibili](#)

[不愧是李宏毅教授, 半天就教会了我 Self-Attention 模型! 自注意力机制和 Transformer 从零解读, 论文解读+源码复现! \(人工智能/深度学习\)_哔哩哔哩_bilibili](#)

(3).对自己研究的细分方向, 找一些经典方法进行深入阅读。

3.代码实践能力:

(1).能够成功运行其其他人的网络模型, 并对自己的图片进行测试

理解 Pytorch 环境配置, conda 环境配置, 并成功安装 pytorch,理解 dataset_loader 代码的基础逻辑

注: conda 环境需要下载 anaconda 软件。创建虚拟环境, 每个环境之间是独立的, 有助于我们针对不同的项目的要求创建不同环境, 安装不同依赖, 方便进行环境的管理。

***注意:** 要熟练掌握环境搭建以及 Python 库版本之间的对应。在使用开源代码时要关注该代码所依赖的 packages。不同的项目底层 packages 可能有不同版本的要求, 通常我们会创建虚拟环境来进行各种 packages 的配置管理, 这时候就要考虑不同版本 packages 相互的兼容性 (例如 pytorch&cuda&python 的版本对应关系)。

一些重要的版本对应关系：

torch	torchvision	Python
main / nightly	main / nightly	>=3.8, <=3.11
2.0	0.15	>=3.8, <=3.11
1.13	0.14	>=3.7.2, <=3.10
1.12	0.13	>=3.7, <=3.10

torch	torchvision	python
main	nightly	>=3.7, <=3.10
1.12.0	0.13.0	>=3.7, <=3.10
1.11.0	0.12.3	>=3.7, <=3.10
1.10.2	0.11.3	>=3.6, <=3.9
1.10.1	0.11.2	>=3.6, <=3.9
1.10.0	0.11.1	>=3.6, <=3.9
1.9.1	0.10.1	>=3.6, <=3.9
1.9.0	0.10.0	>=3.6, <=3.9
1.8.1	0.9.1	>=3.6, <=3.9
1.8.0	0.9.0	>=3.6, <=3.9
1.7.1	0.8.2	>=3.6, <=3.9
1.7.0	0.8.1	>=3.6, <=3.8
1.7.0	0.8.0	>=3.6, <=3.8
1.6.0	0.7.0	>=3.6, <=3.8
1.5.1	0.6.1	>=3.5, <=3.8
1.5.0	0.6.0	>=3.5, <=3.8
1.4.0	0.5.0	==2.7, >=3.5, <=3.7
1.3.1	0.4.2	==2.7, >=3.5, <=3.7

Table 3. CUDA Toolkit and Corresponding Driver Versions

CUDA Toolkit	Toolkit Driver Version	
	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 11.8 GA	>=520.61.05	>=522.06
CUDA 11.7 Update 1	>=515.48.07	>=516.31
CUDA 11.7 GA	>=515.43.04	>=516.01
CUDA 11.6 Update 2	>=510.47.03	>=511.65
CUDA 11.6 Update 1	>=510.47.03	>=511.65
CUDA 11.6 GA	>=510.39.01	>=511.23
CUDA 11.5 Update 2	>=495.29.05	>=496.13
CUDA 11.5 Update 1	>=495.29.05	>=496.13
CUDA 11.5 GA	>=495.29.05	>=496.04
CUDA 11.4 Update 4	>=470.82.01	>=472.50
CUDA 11.4 Update 3	>=470.82.01	>=472.50
CUDA 11.4 Update 2	>=470.57.02	>=471.41
CUDA 11.4 Update 1	>=470.57.02	>=471.41
CUDA 11.4.0 GA	>=470.42.01	>=471.11

CUDA 环境	PyTorch 版本
9.2	0.4.1, 1.2.0, 1.4.0, 1.5.0(1), 1.6.0, 1.7.0(1)
10.0	1.2.0, 1.1.0, 1.0.0(1)
10.1	1.4.0, 1.5.0(1), 1.6.0, 1.7.0(1)
10.2	1.5.0(1), 1.6.0, 1.7.0(1), 1.8.0(1), 1.9.0, 1.9.0, 1.10.0, 1.10.1, 1.11.0, 1.12.0, 1.12.1
11.0	1.7.0(1)
11.1	1.8.0(1), 1.9.0, 1.10.0
11.3	1.8.0(1), 1.9.0, 1.9.1, 1.10.0, 1.10.0, 1.10.1, 1.11.0, 1.12.0, 1.12.1
11.6	1.8.0(1), 1.9.0, 1.10.0, 1.12.0, 1.12.1
11.7	1.12.0, 1.12.1, 1.13.1

(2).能够把论文中的网络模型和给出的复现代码进行一一对应

初步学会看论文的网络图，以及理解 Pytorch 给出的函数接口，理解 Pytorch 如何表示卷积，如何表示池化，对代码复用层面如何处理，对于跳连接级联如何进行处理，理解模型的构建与封装，主要是理解 Model 代码为文件。

难点是对类中的定义以及 forward 实际前馈的理解，需要对 python 类有一定理解。

(3).能够把根据需求去更改网络,对网络结构进行修改

通过实操过程中，对于 feature map 尺寸的变化有一定把握，并且进一步理解 model 的 forward 和 backward 过程。

(4).不仅能够更改网络层数，还能为网络设计新的算法

能够根据利用 pytorch 进行 tensor 矩阵运算，来进行优化算法的设计，部分模型需要有 numpy 的基础。

4.阅读论文的一些误区和方法：

(1).有很多同学英语基础刚开始并不好，对着英文死磕，光是概要的英文已经是大汗淋漓了，然后头脑发晕。可以借助翻译软件查询陌生词汇，对于经典论文可上网查询相

关讲解。

(2).专业词汇读不懂:

例如 `GruondTruth` 这个词，最开始没能理解，翻译软件也翻译不过来，或者是 `convex` 这种优化相关的数学名词，翻译起来会出现仍然看不懂的情况，此处推荐 [wiki 百科](#)。

(3).深度学习领域文章阅读要点

a.可先读概要和结论，来理解背景，提出的方法，实验效果，以及科研展望

b.其次是阅读网络模型框架图，如果遇到看不懂的地方，下方的文字会给出严谨的文字解释，

有看不懂的地方可以到下方文字解释中查阅。

c.作者有几率会在这里提出一些自己造的名词，通过将简写恢复为全称有助于弄懂，了解整个网络的设计思路，并且可以结合代码理解，用代码和网络一一对应。在过程中也可进一步理解 `Pytorch` 编程，学习他人 `pytorch` 编程经验

5.学术能力:

(1) 阅读文献的总结提炼能力:

- 在广泛查找资料文献时可以主要关注文章的摘要，略读方法部分，若觉得有一定的学习价值，则可收藏起来进行深入研究
- 在阅读了一定数量的文献后，建议对具有类似思想的论文可以进行归类整理和对比，归纳当前有哪些主流方法。示例:

姓名: 何吴蓝	时间: 2023.2.12
一、工作内容（所读论文详细介绍；所做实验详细进展；所思考的个人想法；相关思路的研究进展；个人想法的完善程度等）：	
1、梳理了一下这段时间看的 <code>few-shot</code> 论文， <code>matching</code> 的方法可以大致总结如下：	
(1) <code>support prototype</code> 的提取大多都用 <code>MAP</code> (<code>masked average pooling</code>) 来做，还有一些是构建 <code>super-pixel</code> 来做。	
(2) <code>Support prototype</code> 和 <code>query</code> 的 <code>correlation</code>	
A. 余弦相似度	
B. <code>Concatenate</code> 再卷积	
C. <code>GCN</code> 方法	
D. 矩阵变换和相乘来获取关系	
E. <code>Transformer</code>	

- 深入研究一篇文献时可以从以下**几个方面进行总结**（可以梳理在周报中，有助于

后续查阅)：

本文聚焦什么 tasks、相比先前的方法本文解决了什么问题、本文的核心思想（用一句话概括方法）、该方法的理论依据或者灵感来源（是否具有借鉴意义）、本文的主要贡献、提炼方法的主要内容、**该方法的不足之处（进一步：我们可以怎么改进他）**

(2) **英语基础**：在阅读英文文献时尽量少依赖翻译软件，要逐渐习惯英文的学术阅读

(3) **作图**：PPT 作图 yuds（不要用 WPS 的 ppt，导出 PDF 往往有压缩）、WPS 可以绘制流程图、matlab 画数据图（要掌握 matlab 的使用）、小红书或者知乎上可以了解一下高级配色（**论文图表的美观很重要**）

(4) **写作**：掌握 Latex 使用（可以用 <https://www.overleaf.com/>）

二、攻略：

(以下学习攻略同一模块任选其一即可，如果学习中遇到困难则可以在其他视频或书籍寻找答案)

1.学习 Python 及常用库：

学习平台：1. Jupyter notebook 使用简单，功能强大的学习工具

[最易上手的 Python 环境配置——Jupyter Notebook 使用精讲_哔哩哔哩_bilibili](#)

2.实验楼：无需安装配置的学习网站，即开即学

(python、numpy、pandas、Linux....)

[精选项目课程_IT 热门课程_蓝桥云课课程 - 蓝桥云课 \(lanqiao.cn\)](#)

(1)：Python（**建议掌握基本语法后多多实践加深记忆**）：

资料推荐：

视频：

[【Python 教程】《零基础入门学习 Python》_哔哩哔哩_bilibili](#)

[Python 语言程序设计_北京理工大学_中国大学 MOOC\(慕课\) \(icourse163.org\)](#)

书籍：

《python 从入门到实践》

链接：https://pan.baidu.com/s/1JkTK-wfKy-7_zni83FgmFA 密码: juwa

《流畅的 Python》

https://blog.csdn.net/weixin_30263073/article/details/97508102

(2): 各种库函数 (通过项目实践学习各种库函数的使用方法)

★ Numpy、Pandas (数据处理库)

视频 [【莫烦 Python】 Numpy & Pandas \(数据处理教程\)_哔哩哔哩_bilibili](#)

numpy 的矩阵运算、广播机制

Pandas 数据读取与保存, 以及基础的数据处理

实验楼 [NumPy 百题大闯关_Python - 蓝桥云课 \(lanqiao.cn\)](#)

os、pathlib (文件与系统操作)

OpenCV、pillow (图像处理)

matplotlib、seaborn (数据可视化)

(3): Pytorch

保姆级 Pycharm、Jupyter 安装配置和 Pytorch 环境配置以及 Pytorch 学习: (小土堆) 9h

[PyTorch 深度学习快速入门教程【小土堆】_哔哩哔哩_bilibili](#)

TASK1: 你有一个目录, 装了很多照片, 把它们尺寸变成都不大于 iPhone5 分辨率的大小。

并且统计出他们的文件名, 并根据此生成一个 file_names.lst 文件

TASK2: 你有一个目录, 现在需要在里面生成 100 张验证码各不相同的图片。文件名就是他的验证码, 并且生成一个 filename.lst

2.深度学习理论基础 (重点) : 都可在 jupyter 上学习

(1): 深度学习导论

书籍《深入浅出神经网络与深度学习》《深度学习入门: 基于 Python 的理论与实现》

链接: <https://pan.baidu.com/s/13brIFni4wiCSRahL12hEYw> 提取码: 74xn

(2): 深度学习入门（可以书籍和 CS231n 同步看）

书籍 《动手学深度学习-Pytorch 版》

李沐老师的书，B 站有配套课程。在 [jupyter 记事本](#) 上有全书的内容和交互式的学习方法。该书知识全面，包含深度学习基础、经典神经网络、优化算法等知识。可以在学习过某神经网络后去 b 站看李沐老师的论文精讲系列。

链接：<https://pan.baidu.com/s/1gturpzJcLvDy1-vEYwSnMQ> 提取码：mh7I

学习深度学习基础，看懂代码，手抄代码

《解析深度学习 卷积神经网络原理与视觉实践》其难度低于 CS231n，是一本很好的入门书籍。新手入门，可不拘泥于公式推导，而应将重点放在知识点的掌握，公式推导需要一定的数学基础，其中包括高等数学、线性代数、最优化和研究生的部分课程

★ CS231n 李飞飞课程

作业链接 建议边看作业边看视频，带着作业的问题去视频里找方法，视频内容涉及代码比较少，在写作业的过程中通常需要查阅资料，完成 CS31n 的作业可以加深理解、提升代码能力。

视频 B 站（未必要全看完，可结合文本资料重点掌握基本概念）；

已尝试完成 CS231n 的作业，如果不会做可以看一下[这里](#)。（如果觉得李飞飞课程不是很看得懂，可以结合其他视频，如 b 站李宏毅深度学习视频学习某一知识点；或者可以结合李飞飞课程相关笔记来学习，链接如下

<https://zhuatlan.zhihu.com/p/21930884>)

根据作业要求从 0 开始手推公式，手写代码

Task1:训练一个网络识别手写数字

Task2:利用 [Basnet](#) 将 1-Task1or2 中的所有图片转化为显著性图片

Task3:自己寻找一个网络模型，对 1-Task2 的验证码进行预测，并计算准确率

Task4:利用 [Basnet](#) 结合显著性图片实现抠图

Task3 参考资料：https://github.com/braveryCHR/CNN_captcha

Task4 参考资料：<https://www.cnblogs.com/xypbk/p/9116215.html>

备注：3-TASK1,4-TASK1 是必做任务

在任务验收过程中，会针对对应的代码进行延申提问来看个人的掌握情况

3.实验室服务使用：

1、远程连接服务器

工具推荐：MobaXterm 或者 VScode ssh remote

2、Linux 使用

学习平台：[Linux 基础入门_Linux - 蓝桥云课 \(lanqiao.cn\)](https://www.lanqiao.cn)

Linux 命令：<https://www.lanqiao.cn/courses/1>

Linux 命令速查：<https://www.runoob.com/linux/linux-command-manual.html>

跑自己的代码必须在自己创建的环境中跑!!! 创建虚拟环境（根据环境要求配置环境和相应的库版本）参考 <https://zhuanlan.zhihu.com/p/94744929>

常用的部分命令：

- 移动：利用 mv 命令，可能会涉及到通配符例如 mv *.jpg ./target 就是把当前文件里的所有文件移动对应的文件夹里
- 删除：利用 rm 命令，rm *.jpg 可以把当前文件夹中的 jpg 全部删除
- 结束进程：kill -9 [PID] ,暂停进程 Ctrl+Z
- **挂起**（**必用!** 关闭自己的电脑远程服务器的代码运行不会中断）：nohup &
- 查看进程号：nvidia-smi 查看当前运行在显卡钟的程序

基础命令：cd,dir,python xx.py,rm,mv,nvidia-smi

Conda 相关命令：conda create,conda activate,conda env list

4.其他

- Git: 通过 github 平台进行代码管理，且能够实现多版本、多人协作等功能，在未来研究工作中有很大作用。强烈推荐学习，可以通过参与**开源项目**，来锻炼 git 的使用能力。

廖雪峰的教程 (<https://www.liaoxuefeng.com/wiki/896043488029600>)，学习使用 [GitHub](#) 或者 [GitLab](#)

- Matlab: 了解基本语法，提供了丰富的数学计算函数，在需要用的时候查询相关资料使用即可
- Latex: 大致了解如何使用（尤其是图表），没必要系统学，可用 [overleaf](#)
- Markdown: 推荐软件 jupyter、typora

三、学习原则：

1.建议保持至少一半时间在实操，一半时间在学理论知识

2.切记勿要过度深度遍历知识，知识无尽的，如果深度钻研下去，可能没有尽头，最重要的是框架，以及高频次用的知识

对于学到的知识 一定要注重实践 不能只学习理论知识

例如学习 Pytorch，应该尽快理解 Pytorch 编程的框架，dataloader, model, train, test 文件

例如学习学习过程中学习到 Linux 命令，不是需要把整本 Linux 书籍全部看完后才可使用

3.敢于否定，如果对于给出的学习资料看不懂，要尝试自己寻找学习资料，找到适合自己的那份学习资料

4.询问问题的时候，能够提出具体的问题，并能先给出自己的思考，自己做出的努力，以及目前的困惑

例如卷积是什么？

可以这样问应该是说我查了下卷积，说是表示信息融合，表示的是过去的信号能够对现代的信号造成影响，但这和图像领域的卷积我完全发现不了有什么联系？来表示对回答者的尊重

对于：怎么样学习英语，怎么样学习深度学习，或者卷积在 Pytorch 怎么写，这些过细或者过大的问题在自己的思考后，就很难问出来了。

提问的艺术：<https://bbs.csdn.net/topics/390307835>

PS：对于代码的提问，建议截全报错，截全代码，代码的错误一般比较难找，请求他人 debug 是一件极其耗费精力的事情。

5.框架意识，学习任何知识一定要有框架意识，否则容易陷入细节之中，迷茫找不到相关意义和方向。

6.Python 课程并非需全部浏览

AI 的一些常用工具

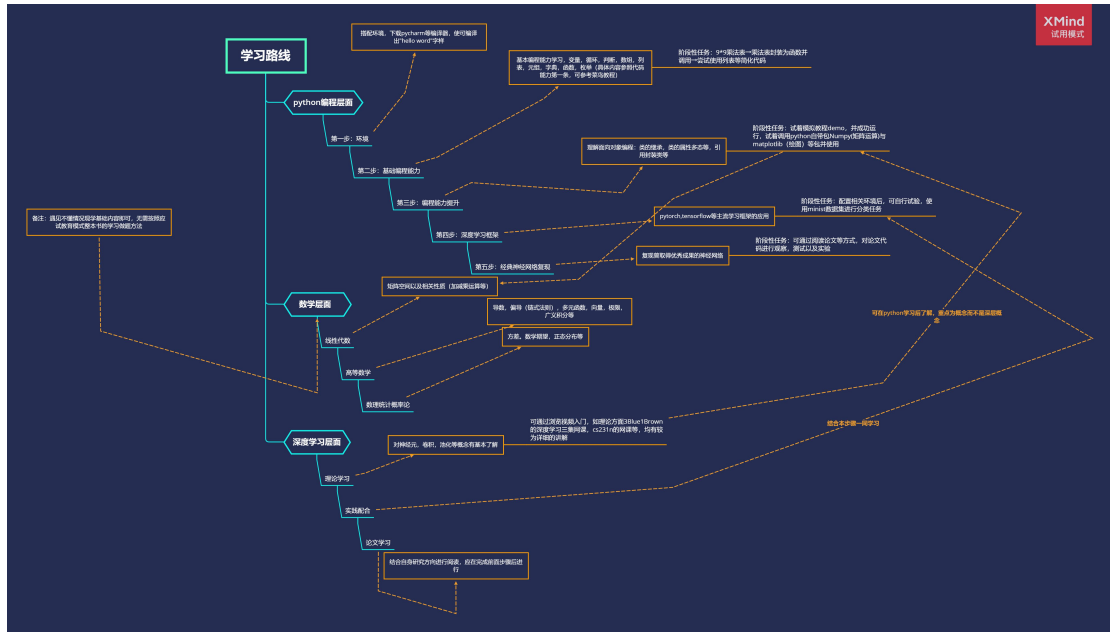
[GitHub - binggoml/AI-research-tools: AI 方向好用的科研工具](#)

常见面试题：[GitHub - NLP-LOVE/ML-NLP: 此项目是机器学习\(Machine Learning\)、深度学习\(Deep Learning\)、NLP 面试中常考到的知识点和代码实现，也是作为一个算法](#)

Pytorch 的最佳实践：

<https://github.com/IgorSusmelj/pytorch-styleguide>

四、知识框架图



Written by Longxuan Yu(于龙轩), Zheming Zhang(张哲铭), Kunye Shen(沈坤烨), Zhicong Wu(吴智聪), Xiaobo Fei(费晓波), Xiaofei Zhou(周晓飞)

Aug.15.2021

Haolan He(何昊蓝) 更新 June.9.2024

Hangzhou Dianzi University