

BIT秋季学期工程实践课程 - 图像处理组项目 实况

本人北京理工大学大三学生，刚刚结束了大三上学期的课程，现在是寒假阶段，对人工智能的目标检测比较感兴趣，但当时工程实践课程被分到了ROS组，故开始将图像处理和目标检测组的实践作业学习，学习过程更新于下。

第一讲：智能感知与信息处理基本概念

任务1：安装Ubuntu操作系统（安装前做好重要文件备份工作）

参考 <https://blog.csdn.net/hwh295/article/details/113409389>，在3.5环节将最小安装改为正常安装，在3.7分区中建议将/boot分区划分2GB或以上空间任务1拓展（可选）：若显卡是NVIDIA品牌，在ubuntu系统上，仔细阅读英伟达官网驱动以及CUDA的说明文档，安装英伟达显卡驱动以及CUDA（安装不当可能出现黑屏、循环登录等情况，谨慎进行）

任务2：在Ubuntu及Windows操作系统下安装anaconda

任务3：在Ubuntu系统下安装PyTorch

任务3拓展（可选）：若完成任务1拓展，强烈建议安装GPU版本的PyTorch重要提示：若任务1拓展未完成，则在Windows系统下安装英伟达驱动与CUDA，并按照指南安装PyTorch的GPU版。也就是说，在Ubuntu或者Windows系统下，只要是N卡，必须有一个GPU版的PyTorch

任务4：在Ubuntu系统下安装ROS

第二讲：信息处理基础及实践

任务1：利用jupyter notebook运行梯度下降以及逻辑回归程序，理解其中每一段代码

拓展任务（可选）：下载波士顿房价数据集，在网上查阅相关资料，试利用回归模型对数据集进行回归预测，并评价模型的预测效果，同时对数据网站Kaggle进行了解，数据集地址：链接：https://pan.baidu.com/s/1vTWjvMjl1bTDCmoka_msw 提取码：y7km

or <https://www.kaggle.com/>（kaggle网站，在其中搜索Boston）

拓展任务（可选）：自行搜索Iris鸢尾花数据集，结合相关资料，选取合适的模型完成鸢尾花数据集分类任务

任务2：利用PyTorch编程搭建一个全连接神经网络，该神经网络具体结构如下：

接收一个维度为 1×2 的向量作为输入；具有一个隐藏层，隐藏层包含3个神经元；输出层输出一个维度为 1×2 的向量；任选激活函数；可指定权重参数，也可随机初始化

**** 要求：**1. 任选三组输入，将其输入上述神经网络，观察其输出，判断输出是否与理论计算相符合
2. 改变激活函数，比较不同激活函数下神经网络输出差异**

提示：参考链接：链接: https://pan.baidu.com/s/1aGEP_mIPNAzKMA1LLMcW6A 提取码: f2ic

任务3：利用PyTorch编写程序，训练一个全连接神经网络，完成MNIST手写数字识别任务

要求：简单说明训练神经网络的结构，并绘制训练过程中的损失函数曲线（提示：在www.github.com上搜索相关资源，利用matplotlib软件包绘制损失函数曲线）

拓展任务（可选）：自己生成一组有规律的回归数据以及分类数据，将这两组数据分别制作数据集，并设计相应的神经网络模型对数据进行分类和回归，观察两类模型哪一类的训练效果更好。

提示：制作数据集可以使用torch.utils.data.Dataset类，并在其中重新定义len()以及getitem()两个函数

第三讲：图像识别与卷积神经网络

任务1：利用PyTorch编写程序，训练一个卷积神经网络，完成MNIST手写数字识别任务

提示：代码链接： <https://pan.baidu.com/s/1ZoaXSx8Gf0b30N3MTL93EA> 提取码: tnpj

拓展任务：登陆PyTorch官网及相关网站，了解FashionMNIST数据集，并训练一个卷积神经网络完成FashionMNIST的训练

任务2：熟悉经典图像分类网络

要求：阅读VGG以及RESNET论文原文，了解其设计思路，按照论文对网络模型进行复现。找到PyTorch关于VGG16，ResNet两个经典图像分类网络的官方实现，进行比对。

拓展任务：尝试利用VGG模型对CIFAR10进行训练（训练结果不作要求）

第四讲：卷积神经网络进阶：目标检测

任务1：自己动手实现NMS算法，自行生成一组矩形框数据并验证算法效果

任务2：结合资料与YOLOv3论文原文对YOLOv3进行进一步了解，对其训练和推理过程形成清晰认识

任务3: 在 <https://github.com/ultralytics/yolov3> 下载 YOLOv3, 运行其tiny版本, 比较其与正常版本在速度与准确率上的直观区别

(提示: 在Windows环境下, CUDA容易安装, 可在此基础上安装OpenCV-Python, 打开笔记本电脑前置摄像头, 分别运行YOLOv3和YOLOv3 Tiny)

第五讲: 机器人操作系统与点云信息处理

任务1: 将listener.py和talker.py改写成cpp格式, 修改相应的CMakeLists.txt文件与package.xml文件, 令两个程序之间能够成功通信

拓展任务(可选): 使用talker.cpp和listener.py进行通信, 学习使用launch文件, 尝试同时启动两个节点

任务2: 编写程序, 尝试将点云三维框信息投射到二维图像信息上

(提示: 对应资源: kitti部分数据集链接: https://pan.baidu.com/s/1rcJkYl5ap73H_p-k3O_XGA 提取码: iyqb)

结课作业

1. 利用COCO128, 完成对YOLOv5的训练。项目地址: <https://github.com/ultralytics/yolov5>, 如无GPU或GPU训练速度慢, 则熟练掌握PyTorch的dataset类和DataLoader类, 制作COCO128的数据集, 要求: 若数据集实例命名为coco, 则iter(coco).next()能够返回对应一组图像序列及对应标签。

2. 修改YOLOv5源码, 录制一段视频, 令其只框出人类目标 两种方法可选:

(1) 简单修改代码, 在显示阶段抑制其他目标输出 (2) 自行寻找行人数据集, 对YOLOv5重新训练 GPU受限则自行选取一组含行人的照片 也可发挥主观能动性, 不受题目限制, 用任意检测模型实现其他检测功能