

《Python 人工智能应用》作业说明

《程序设计实践》作业 3

2021 年 07 月

一、作业内容

本次作业内容比较简单,主要目的是让同学们了解并掌握如何使用 Python 进行人工智能模型的训练和推理。

考虑到并非所有同学都有深度学习方向的背景知识,本次作业将提供基础的代码框架,同学们只需要根据任务说明,掌握示例代码各个步骤的含义,对其中的小细节进行修改即可。

二、任务说明

作业资料提供了针对数字手写体识别任务的训练代码和推理代码,同学们需要尝试运行代码,并完成以下小任务:

- 1、使用数据集中的训练数据,拼出自己的学号;
- 2、训练一个模型,并使用它进行推理;
- 3、绘制模型训练的 loss 随 epoch 变化的曲线图;
- 4、尝试更换优化器,查看训练效果是否有变化;
- 5、添加数据预处理;
- 6、将训练完成的模型保存成文件,并通过可视化工具显示模型的结构;
- 7、错误样例分析;

大多数任务都可以在现有代码的基础上快速完成,主要目的是希望大家理解使用 Python 进行人工智能模型训练的过程。

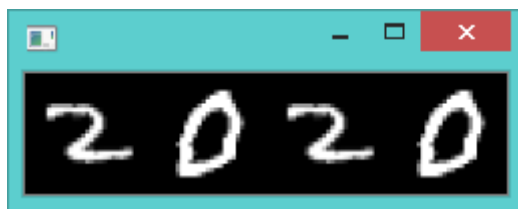
本次作业需要同学们在 python 环境中安装 Pytorch 和 opencv 两个包。

Pytorch 安装方法: <https://pytorch.org/get-started/locally/>

Opencv 安装方法: `pip install opencv-python`

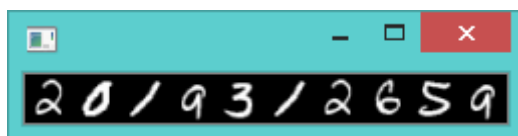
2.1 图片可视化

执行 `task-1.py` 文件,会输出从数据集中选取的图片拼接而成的图片组合。



生成图片组合的具体逻辑可以在 ``utils/visualizer.py`` 文件中给定的方法 ``demo_display_specific_digit_combination`` 中找到。

请阅读并参考示例代码，输出用 MNIST 数据集中一个批次图片拼接出的自己学号，截图后放在作业文档中（如下图所示）：



2.2 训练模型并使用它进行推理

运行 ``train.py``，完成模型的训练和保存，然后运行 ``inference.py``，完成模型的推理过程。上述文件中的代码在不经修改的情况下应该可以直接运行，只需要根据实际场景修改输入参数即可。

在 ``inference.py`` 文件中默认已经添加了显示图片的逻辑，在完成模型的推理过程后，将自动将预测结果绘制在图片上并显示图片。请同学们使用训练好的模型，针对自己学号最后一位数字的图片进行推理操作。完成这个流程后，将运行结果记录在作业文档中。（如下图所示）



注意：如果模型预测的结果与实际数值不符也没有关系，因为仅在 MNIST 数据上训练的数字识别模型对于其它图片的泛化性有可能不够好。

2.3 绘制 Loss 曲线

在训练过程中，将每个 epoch 的 loss 记录用于统计，在训练完成时绘制 Loss 曲线，可以调用任意可视化算法包。

请在文档中放置绘制的 Loss 曲线截图，并说明绘制图像使用的方式（及使用的第三方库，如 matplotlib 等）。

2.4 更换优化器

修改 `train.py` 训练相关代码，将优化器 Adam 更换为 SGD，观察输出结果。该任务只需要将修改优化器的那行代码（修改后的代码）粘贴到文档中即可。

2.5 添加数据预处理

尝试在数据预处理的步骤中添加一些步骤。代码 `utils/pre_process.py` 中定义了两个基本的数据变换方法，请填写 `data_augment_transform` 方法，分别添加下面几个数据增广的操作：

- 随机剪裁
- 水平翻转
- 垂直翻转

在 `torchvision` 包中提供了许多数据预处理方法，可以参考官方文档进行修改：<https://pytorch.org/docs/stable/torchvision/transforms.html>。

在文档中说明添加数据预处理的方式，以及每一种数据增广操作添加后对于模型的训练效果有何影响。

2.6 模型导出及模型可视化

尝试修改 `models/lenet.py` 文件，在模型中添加一层卷积层，进行训练，并使用导出示例代码，导出模型文件，使用任意一款模型结构可视化工具查看其模型结构。

将模型结构的截图放置在文档中即可。

提示：

1、添加的卷积层中 `input_channel` 和 `output_channel` 要确保与前一层网络和后一层网络相符，此外 `kernel_size`、`stride` 和 `padding` 会对输入的形状有影响。

2、模型结构可视化可参考 <https://lutzroeder.github.io/netron/>。

3、如果对深度学习相关知识不够熟悉的同学，可以联系助教解决。（这不是一个很值得研究的问题，避免在此花费超过 30 分钟。）

2.7 错误样例分析

修改评估相关的代码，输出错误情况的输入图片及类标的信息。在文档中输出其中 5 个模型推理错误的情况，试分析模型判断错误的可能原因。

提示：在 `train.py` 文件中的 `evaluate` 方法中，对模型输出结果和真实标签进行对比。

三、提交格式

本次作业最终只需要提交完成后代码文件以及作业文档，请按照以下格式组织提交的文件：

- [学号]_[姓名]_hw1
 - src
 - doc

即创建一个名称为 `[学号]_[姓名]_hw3` 的文件夹，放置 `src` 目录和 `doc` 文目录，分别放置代码文件和作业文档。注意代码文件只要上传后缀为 `.py` 的文件，不要上传数据和模型。比如对于学号为 `2030010001`，姓名为 `张三` 的同学，文件夹的名称应该为 `2030010001_张三_hw3`（注意使用下划线而非其它字符拼接学号和姓名）。将整个文件夹打包成 `.zip` 压缩包后，上传到网络学堂。

作业文件中提供了示例版本的作业提交格式。注意，文件夹中的学号和姓名

将直接用于统计作业分数，未按照要求格式命名的作业将无法被代码正确解析，会酌情扣分。

四、评测方式

本次作业主要以熟悉 Python 在人工智能领域的研究方法为主，根据上述任务任务的完成情况进行评测。

评价指标	比例	备注
图片可视化	10%	
完成模型训练和推理	15%	
绘制 Loss 曲线	15%	
更换优化器	10%	
添加数据预处理	10%	
模型导出及可视化	20%	
错误样例分析	20%	

五、其它

注意事项：

- 批改环境将使用 Python3.8 环境，包含 PyTorch 生态的相关库（建议使用 1.1 版本及以上的 Pytorch），如果提交的代码中需要额外安装库，请在 README.md 文件中注明；
- 提交的文件夹和文件名称按照上述要求组织（重要）；
- 参考任务要求在文档中分点撰写自己的实现方案、结果、分析，附带必要的截图和相关代码示例，在实现过程中尽量用一个独立的函数完成特定功能的实现。批改时将根据文档结合代码给分；

希望同学们认真按照作业说明和要求完成作业的提交，避免由于提交格式原因丢失分数。如果有问题或疑问及时与助教(郝天翔, htx20@mails.tsinghua.edu.cn)联系，谢谢！