

课后作业

手写或打印一页A4纸/题

课堂提交

作业编号 5
满分10+5分

Q1 (5分) 实现 2-layer CNN 用于MNIST分类

Step 1: 下载[数据代码](#) MNIST_CNN文件夹，其中MNIST数据整理成npz文件，每一条数据 784维度。原始数据一张MNIST图片大小为 28x28，可以通过reshape变回原来尺寸。

Step 2: 使用 [PIL](#) 和 [scipy](#) 等，可视化其中任意一张图片。[参考代码](#)。

- Hint: 可在 main.py load_mnist_data 中，对 train_images[0,:] 提取第一张 768 维度数据，调用 numpy reshape 为 (28,28) 。再使用 scipy toimage 函数转为 PIL 图像类型，使用PIL save 函数存为 vis.png 进行可视化。

Step 3: 修改 model_cnn_TODO.py 。完成一个 **2-层** CNN 网络用于 MNIST 数据10分类。

- 使用给定的组件，稍加修改其中的必要参数，使得 CNN,ReLU, Dropout, Pooling 等层正确进行组合。可参考[AlexNet](#) 的组合方式。
- 注意最后一层或者两层全连接层的参数值选取，保证正确运行。最终的输出应当为(Batch_size, 10) 的维度的logits，作为 CrossEntropy Loss的输入。参考 train() 函数。

Step 4: 执行 python main.py --type=SGD/ADAM 查看在MNIST上train,val, test的结果。本次实验设置 epoch=8，在CPU上运行时间大约五分钟，GPU上约为1分钟。Mac可以尝试使用MPS设置。

Step 5: 简要解释 TODO-Code, TODO-Explain 部分的代码。比较两种Optimizer的效果。

Q2 (5分) 运行Stable Diffusion

Step 1: 运行Diffusion_UNET文件夹test_unet.py

Step 2: 使用Colab或者AutoDL，得到显存8G以上的GPU

Step 3: [下载](#)模型，在anaconda环境，安装diffuser等。

Step 4: 运行查看输出的 fantasy_landscape.png

Step 5: 查看有多少CrossAttnDownBlock2D和CrossAttnUpBlock2D 层

Bonus (+5分): 查看 UNetMidBlock2DCrossAttn，对照 [源代码1](#) [源代码2](#) 解释作用。可手绘大致unet图形进行解释。