

# 北京邮电大学 2025—2026 学年第一学期

## 《神经网络与深度学习》课程实验作业（三）

### 注意事项：

- ① 本次实验包含两道题，共计 30 分；
- ② 所有实验结果需以实验报告的形式进行提交，文件命名格式：实验三\_姓名\_学号.docx，文件中需要将作者设置为本人姓名；
- ③ 实验报告中需要插入代码片段，完整代码无需放在实验报告中，以压缩包的形式添加即可，压缩包命名格式：实验三\_姓名\_学号.zip；
- ④ 作业提交截止时间：2025 年 12 月 31 日晚上 23: 59

### 1. 基于 MNIST 数据集的自编码器实现 (15 分)

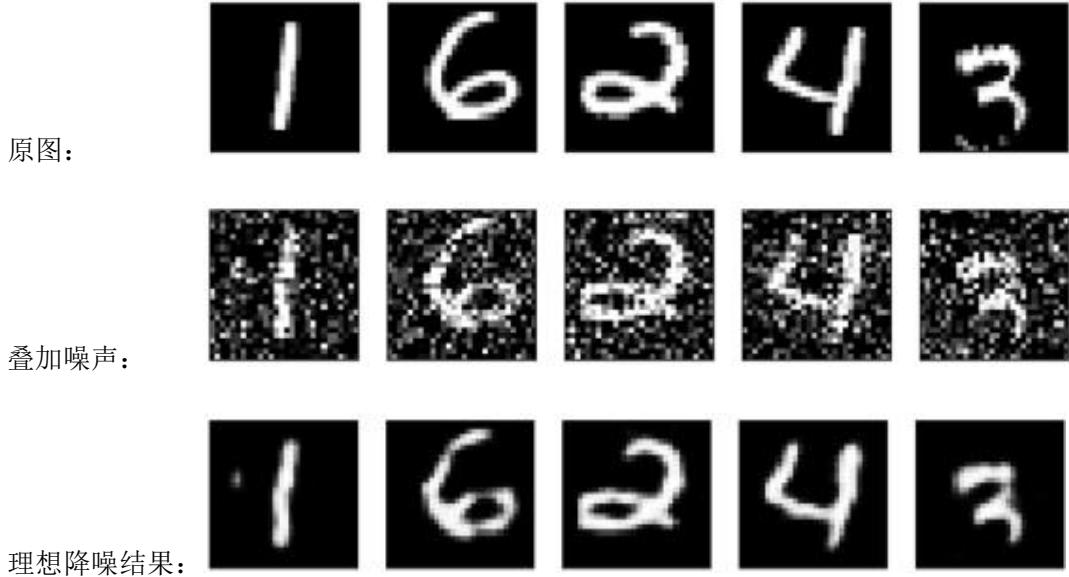
MNIST 数据集来自美国国家标准与技术研究所，National Institute of Standards and Technology (NIST)。训练集 (training set) 由来自 250 个不同人手写的数字构成。请基于该数据集，并结合所学知识完成以下实验内容：

- (1) 完成数据读写并试着搭建深度自编码器网络。使用 print 打印网络结构并将结果截图放入实验报告中。(1 分)
- (2) 选择二元交叉熵函数作为损失函数，在限制 bottleneck 层维度为 2 的情况下训练模型。给出相应代码截图，并给出数字从 0 到 9 的 10 张图片的原始图片和重建图片。(3 分)
- (3) 设置噪声因子为 0.4，在输入图像上叠加均值为 0 且方差为 1 的标准高斯白噪声，训练降噪自编码器，给出相应的代码截图和关键部分代码说明 (4 分)，并进行降噪结果展示，给出数字从 0 到 9 的 10 张图片的原始图片、加噪图片和重建图片。(3 分)。

#### ❖ 代码示例：

```
noise_factor = 0.4
x_train_noisy = x_train + noise_factor * np.random.normal(loc=0.0, scale=1.0,
size=x_train.shape)
```

#### ❖ 降噪结果展示示例：



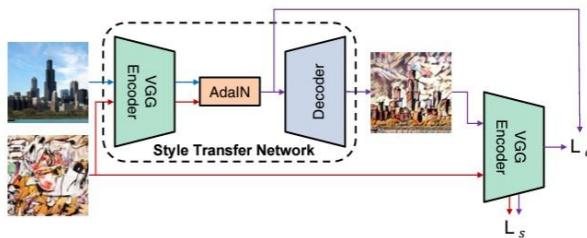
(4) 试在问题(2)的基础上, 对 latent code 进行均匀采样, 并利用解码器对采样结果进行恢复, 展示 latent code 的分布、采样范围和采样结果, 观察并描述所得到的结果。(4 分)

MNIST 数据集下载: <http://yann.lecun.com/exdb/mnist/>

## 2. 图像风格迁移 (15 分)

图像风格迁移任务是将一幅图像的风格应用到另一幅图像的内容上, 从而生成具有目标风格特征的新图像。在过去的探索中, 人们发现图像风格与 latent code 的统计特性高度相关, 修改 latent code 的统计特性可以实现风格迁移。一种简单但高效的方法是基于 Instance Normalization(IN)实现, 在下图中体现为 AdaIN 模块。AdaIN 模块同时接收 content 的 latent code  $x$  和 style 的 latent code  $y$ , 然后基于下述公式对齐均值和方差:

$$\text{AdaIN}(x, y) = \sigma(y) \left( \frac{x - \mu(x)}{\sigma(x)} \right)$$



对 latent code 的统计特性进行修改后, 将其结果输入给 decoder, 即可重建出风格迁移后的图像。后续问题是如何在没有风格迁移结果的条件下进行自监督学习? 答案是再经过一遍编码器, 用 latent code 构建自监督学习。记 content 图像为  $c$ , style 图像为  $s$ , 预训练的 VGG Encoder 为  $f$ , 则风格迁移后的 latent code 为  $t = \text{AdaIN}(f(c), f(s))$ 。使用一个随机初始化的 Decoder  $g$  将 latent code 映射为图像, 生成的风格迁移图像

$$T(\mathbf{c}, \mathbf{s}) = g(\mathbf{t})$$

用预训练的 VGG Encoder 去定义训练 Decoder 的损失函数:  $\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_s$ , 其中  $\lambda$  是超参数用于平衡, content loss 为  $\mathcal{L}_c = \|f(g(\mathbf{t})) - f(g(\mathbf{s}))\|_2$ , style loss 为

$$\mathcal{L}_s = \sum_{i=1}^L \|\mu(\phi_i(g(\mathbf{t}))) - \mu(\phi_i(\mathbf{s}))\|_2 + \sum_{i=1}^L \|\sigma(\phi_i(g(\mathbf{t}))) - \sigma(\phi_i(\mathbf{s}))\|_2$$

其中  $\phi_i$  是预训练的 VGG Encoder 网络的一层, 即 style loss 是基于 VGG 多层输出计算得到的, 更多内容请参考[1]。

MS-COCO (Microsoft Common Objects in Context)是一个广泛应用于计算机视觉的目标检测数据集, 本次实验选取其中的 train2014 作为 content 图像的训练集。WikiArt 数据集是 kaggle 上开源数据集, 用于 style 图像的训练集。每个数据集大约包含 80,000 张图像。试基于上述两个数据集, 完成以下实验内容:

(1) 基于开源项目 <https://github.com/naoto0804/pytorch-AdaIN.git> 训练自己的风格迁移模型, 使用 tensorboard 可视化训练过程, 请将训练过程的结果放入实验报告中。((1)和(2)总共只需训练一次, 但注意保存训练过程中不同阶段的模型) (2 分)

(2) 训练模型的超参数可自行调整, 但总迭代次数不得少于 10000 次。给出迭代次数分别达到总迭代次数的 10%、50%、80% 和 100% 时的风格迁移图像结果。具体要求是, content 图片使用(1)中开源项目的 input/content/cornell.jpg, style 图片使用(1)中开源项目的 input/style/woman\_with\_hat\_matisse.jpg。 (4 分)

(3) 请大家任选北邮的 2 个特色景点, 分别拍取自己和北邮景点的合照, 得到两张不同景点的图像。选取一种风格, 对两张图像进行风格迁移, 展示输入的 content 图像和 style 图像以及输出的风格迁移图像。 (6 分)

(4) 任意选取上述(3)中一张合照, 并使用和(3)中不同的图像风格, 赋予 AdaIN 的输出权重  $\alpha$ , 并赋予 content 图像通过 VGG Encoder 后的输出权重  $1 - \alpha$ , 对二者加权求和后送入 decoder, 设置不同  $\alpha$ , 展示 style 不同占比情况下的风格迁移结果。学号尾号为偶数的同学  $\alpha$  分别设置为 0.3, 0.6, 0.9, 学号尾号为奇数的同学  $\alpha$  分别设置为 0.2, 0.5, 0.8。 (3 分)

参考文献:

- [1] Xun Huang, Serge Belongie, “Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization.” *Proceedings of the IEEE international conference on computer vision*, pp. 1501-1510. 2017.