

北京邮电大学 2023—2024 学年第一学期

《神经网络与深度学习》课程实验作业（三）

实验内容：深度自编码器

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 实验报告中可插入代码片段，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：实验三代码\_姓名\_学号.zip；
- ④ 作业提交截止时间：2023 年 12 月 4 日晚上 20: 00

1. 基于 MNIST 数据集的自编码器实现 (13 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。(2 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。(2 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 (3 分)，并进行降噪结果展示 (2 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：



原图：

叠加噪声：



理想降噪结果：



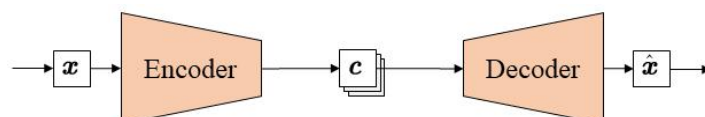
(4) 试在问题(2)的基础上，对 latent code 进行均匀采样，并利用解码器对采样结果进行恢复，观察并描述所得到的结果。(2 分)

(5) 试在问题(4)的基础上，在训练深度自编码器时使用 L2 正则化，观察并描述你所得到的结果。(2 分)

## 2. 隐空间特性探究 (17 分)

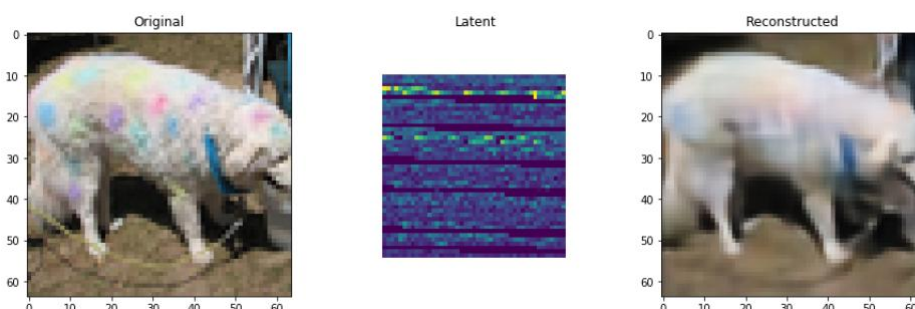
Dog 数据集提供了 2 万多张狗有关的大小各异的图片。试基于该数据集，完成以下实验内容：

(1) 若记输入图像为  $\mathbf{x}$ ，则  $\mathbf{c}$  和  $\hat{\mathbf{x}}$  分别表示由 encoder 编码得到的 latent code 和由 decoder 重建得到的输出图像。请以下图结构为参考，以 MSE 作为损失函数，设置  $\mathbf{c}$  的维度为  $8 \times 8 \times 16$ ，搭建并训练深度自编码器网络。(3 分)



(2) 随机选取 9 张图片，分别展示每一张图片的原图和重建图像，并对 latent code 进行可视化。(2 分)

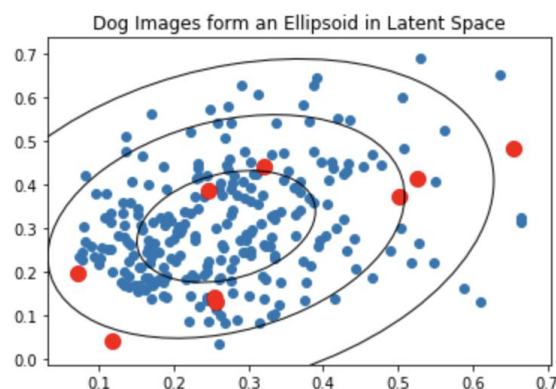
❖ 可视化示例：



(3) 随机选取 256 张图片，通过所构造的自编码器网络中的 encoder 得到其对应的 latent code。计算这些 latent code 的统计特性，并以此为参数构造高斯分布。试在你所得到

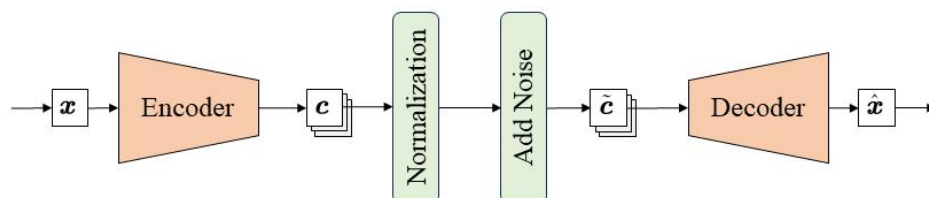
的高斯分布上进行 9 次随机采样，再将采样得到的 9 组 latent code 送入 decoder，观察得到的图像并描述你观察到的现象。(2 分)

❖ 采样示例：



(4) 在任务(3)的基础上，在这 9 张图片的 latent code 上叠加随机的高斯噪声扰动，观察叠加噪声后的 latent code 送入 decoder 生成的图像，并解释你观察到的现象。(2 分)

(5) 如下图所示，请将 latent code 叠加零均值高斯噪声作为一类正则自编码器方法，由此带噪训练新的正则自编码器（限制 latent code 维度为  $8 \times 8 \times 16$ ）。需要注意的是，为了保证高斯噪声具有稳定的效果，还需要在叠加噪声前对 latent code 进行功率归一化。请在噪声方差分别为 0.05, 0.1, 0.15 时，给出 Dog 数据集上重建图像 PSNR 的平均值 (6 分)，需要并探究此时从 latent space 采样是否有生成效果 (2 分)。



提示：

1. MNIST 数据集和 Dog 数据集已在实验三的相关材料包中给出；

2. 可用函数：np.random.multivariate\_normal();

3. torch.normal 函数中的输入参数为标准差，而非方差；

4. 参考公式： $MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|x_{i,j} - \hat{x}_{ij}\|^2$ ， $PSNR = 10 \cdot \log_{10} \left( \frac{\max(\mathbf{x})^2}{MSE} \right)$ ;

5. PSNR 一般不直接作为网络训练时的损失函数，而给出 PSNR 时应给出数据集上的平均 PSNR；

6. 功率归一化 (add\_noise.py) 及损失计算 (distortion.py) 相关代码已在实验三的相关材料

包中给出。