



神经网络与深度学习课程组  
2025-2026 学年第一学期

# 神经网络与深度学习

戴金晟 ([daijincheng@bupt.edu.cn](mailto:daijincheng@bupt.edu.cn))

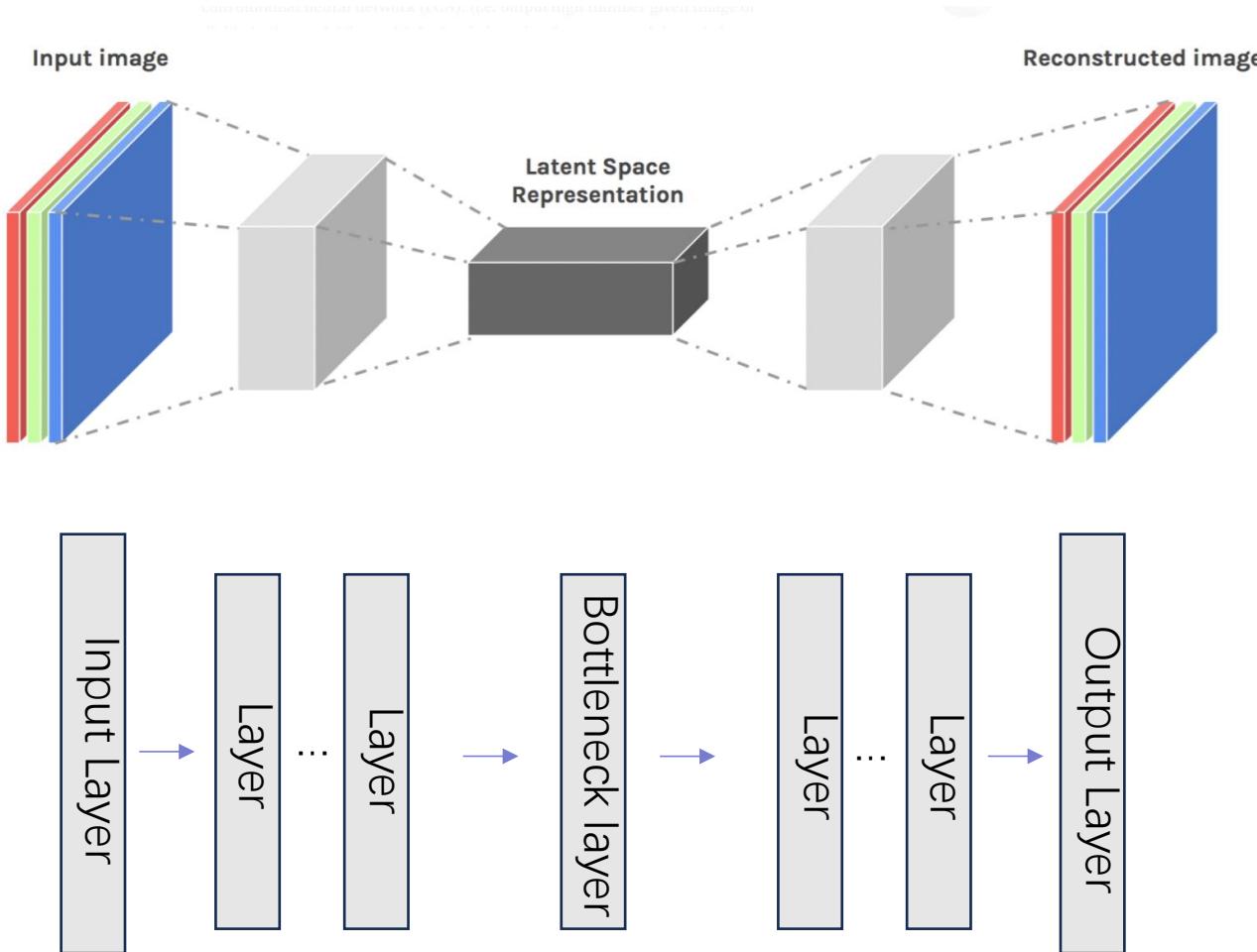
# 实验作业（三）

# 目录

- 一、基于MNIST数据集的AE实现
- 二、图像风格迁移

# 一、基于MNIST数据集的AE实现

- (1) 完成数据读写并搭建深度自编码器网络
- (2) 选择二元交叉熵作为损失函数，限制bottleneck层维度为2，训练模型



```
# ENCODER
input_img = Input(shape=(64, 64, 3))
x = Conv2D(48, (3, 3), activation='relu', padding='same')(input_img)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(96, (3, 3), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(192, (3, 3), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)
encoded = Conv2D(32, (1, 1), activation='relu', padding='same')(x)

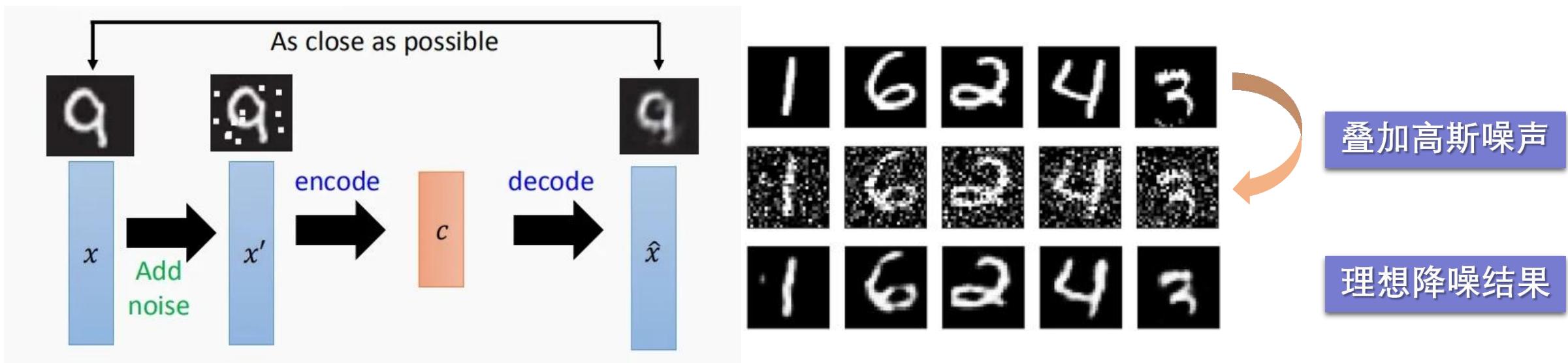
# DECODER
direct_input = Input(shape=latentSize)
x = Conv2D(192, (1, 1), activation='relu', padding='same')(direct_input)
x = UpSampling2D((2, 2))(x)
x = Conv2D(192, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(96, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(48, (3, 3), activation='relu', padding='same')(x)
decoded = Conv2D(3, (3, 3), activation='sigmoid', padding='same')(x)

# COMPILE
encoder = Model(input_img, encoded)
decoder = Model(direct_input, decoded)
autoencoder = Model(input_img, decoder(encoded))

autoencoder.compile(optimizer='Adam', loss='binary_crossentropy')
```

# 一、基于MNIST数据集的AE实现

(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，训练降噪自编码器并展示降噪结果，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

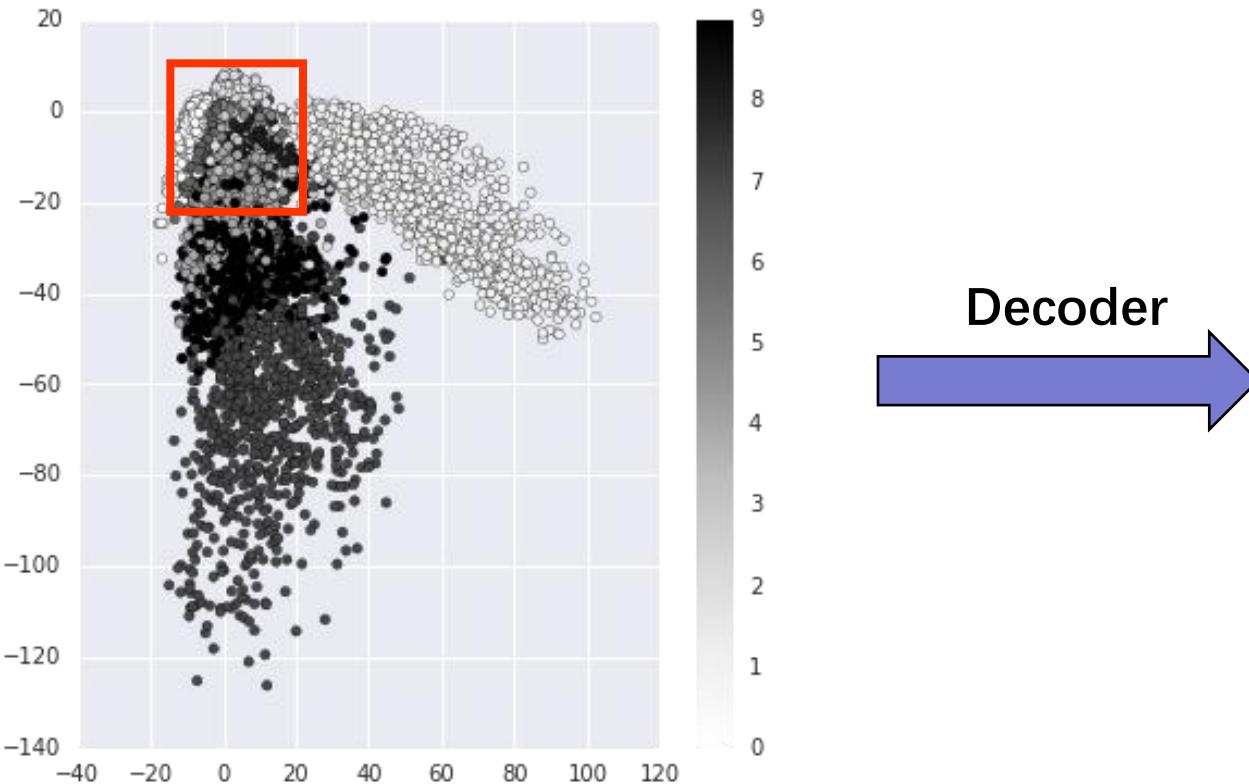


```
noise_factor = 0.4
```

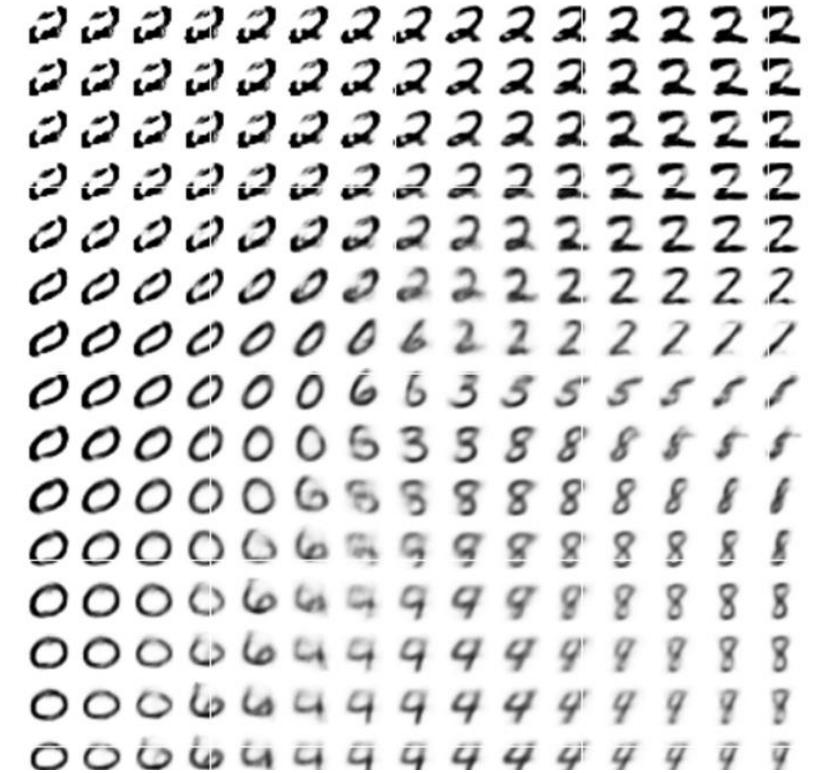
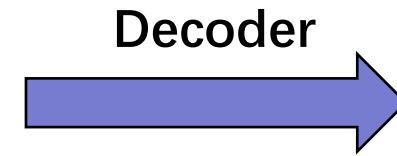
```
x_train_noisy = x_train + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_train.shape)
```

# 一、基于MNIST数据集的AE实现

## (4) 对Latent code进行均匀采样，利用解码器对采样结果进行恢复



latent space中的数字聚类情况



使用解码器将latent space中的采样  
数字重构为与原始输入具有相同维  
度的二维图像

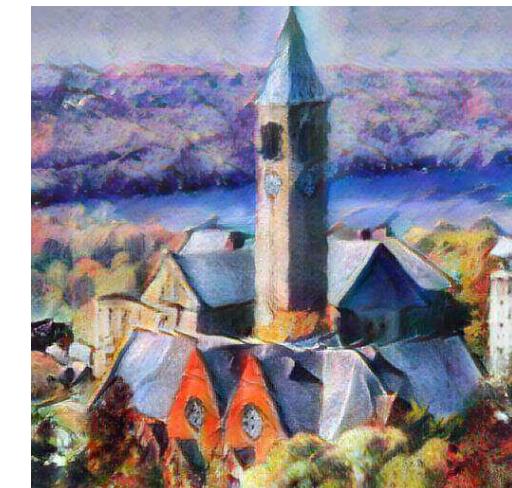
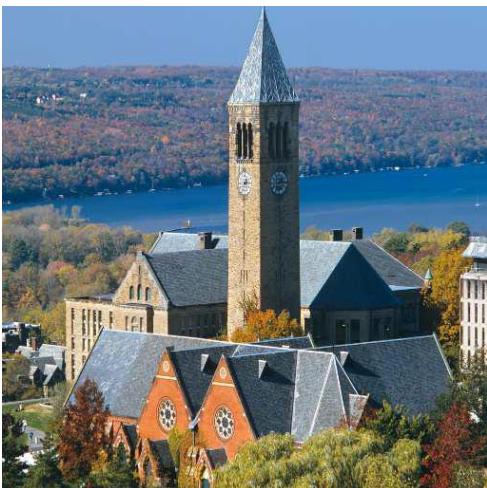
## 二、图像风格迁移

### □ MS-COCO数据集 (content trainset)

➤ MS-COCO数据集是微软构建的目标检测数据集。

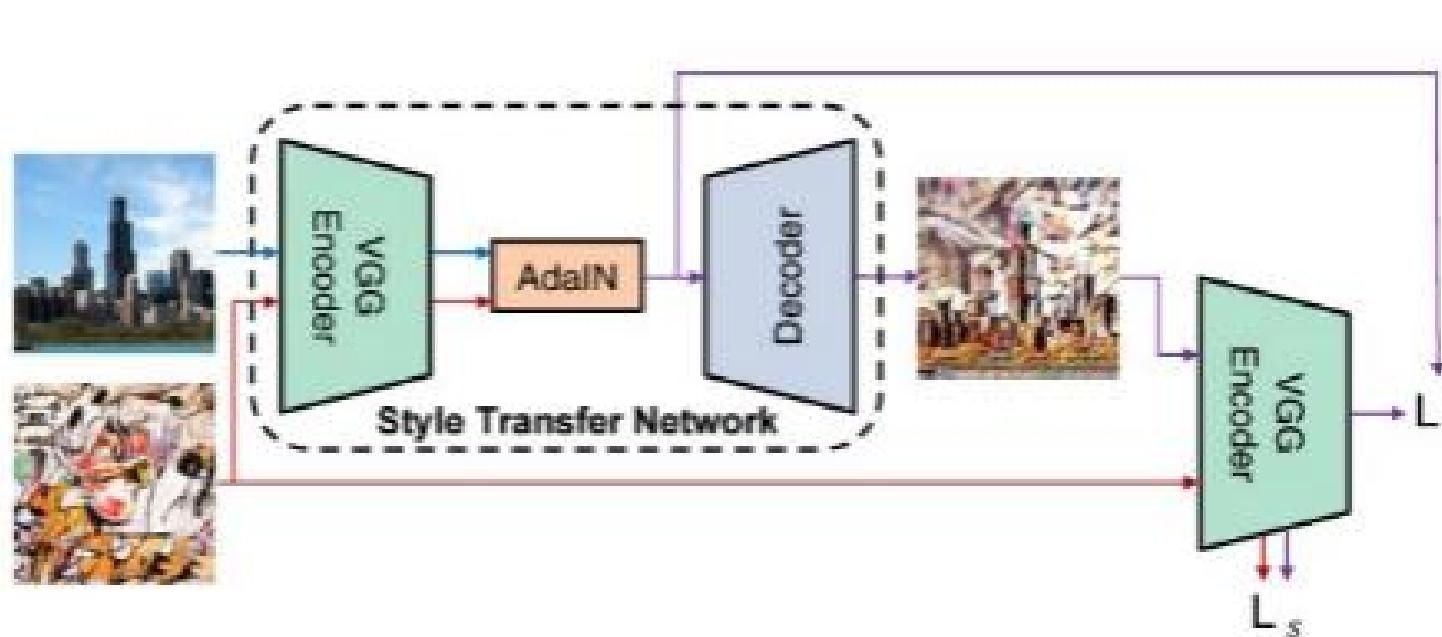
### □ WikiArt数据集 (style trainset)

➤ WiKiArt数据集收录了多位艺术家不同风格作品



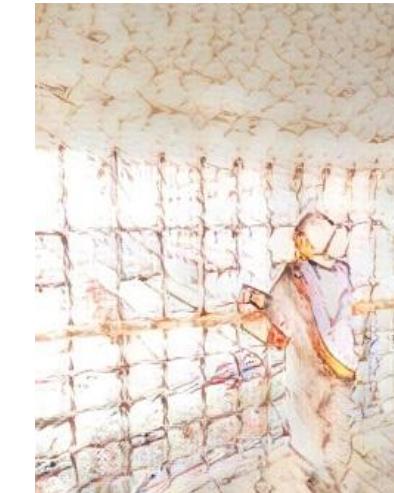
## 二、图像风格迁移

- **任务1：**基于开源项目<https://github.com/naoto0804/pytorch-AdaIN.git>复现模型，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)
- **任务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。



## 二、图像风格迁移

- **任务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。
- **任务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。



## 口如何后台运行程序

➤ nohup命令：

■ nohup python -u train\_dog.py --snr 19 > train\_19\_2.log 2>&1 &

➤ 用jupyter notebook运行后，直接叉掉窗口

## 口如何稳定会话（用于解决kill掉进程的问题）

➤ 用tumx命令，自行查找使用方法

## 注意事项

- 所有实验结果需以实验报告的形式进行提交，实验报告中插入代码片段辅助说明，**实验报告命名格式：实验三\_姓名\_学号.pdf**
- 完整代码和实验报告打包进一份压缩包提交，**压缩包命名格式：实验三\_姓名\_学号.zip**
- 提交至云平台
- 提交截止时间：2025年12月31日晚上24: 00前