

计算机科学与技术学院神经网络与深度学习课程实验报告

实验题目：Style Transfer		学号：201900130151
日期：2021.11.9	班级：人工智能	姓名：莫甫龙
Email：m1533979510@163.com		
实验目的： 取两张图片，把一张图片的风格和另一张图片的内容合成为一张新的图片。		
实验软件和硬件环境： Vs code Win11		
实验原理和方法： content loss用来计算原图片和生成的图片之间像素的差距，这里用的是卷积层获取的feature map之间的差距。 $L_c = w_c \times \sum_{i,j} (F_{ij}^\ell - P_{ij}^\ell)^2$ 公式为: STYLE LOSS 这里我们使用格拉姆矩阵（Gram matrix G）来表示feature map每个通道（channel）之间的联系（也就是风格）。 $G_{ij}^\ell = \sum_k F_{ik}^\ell F_{jk}^\ell$ 输入F的维度为（N,C,H,W），转换为（N,C,M），M=H*W，则输出的G维度为（N,C,C） 则风格的loss就是原图片和生成的图片之间格拉姆矩阵的差距： $L_s^\ell = w_\ell \sum_{i,j} \left(G_{ij}^\ell - A_{ij}^\ell \right)^2$ total variation loss可以使图像变得平滑。信号处理中，总变差去噪，也称为总变差正则化，是最常用于数字图像处理的过程，其在噪声去除中具有应用。		

接下来就是进行完整的风格迁移过程：

初始化img图片（可以初始化为随机噪声或从另一张图片复制）

for t in （迭代次数）

 使用cnn获取feature map

 计算img的总loss（content loss+style loss+tv loss）

 反向传播计算img的梯度

 更新img

实验步骤：（不要求罗列完整源代码）

content_loss:

```
- scalar content loss
"""
# *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
N,C,H,W = content_current.shape
Fc = content_current.view(C,H*W)
Pc = content_original.view(C,H*W)
Lc = content_weight * (Fc - Pc).pow(2).sum()
return Lc

# *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
```

gram_matrix:

```
# *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
N,C,H,W = features.shape
F = features.view(N,C,H*W) # N*C*M
F_t = F.permute(0,2,1) # N*M*C
gram = torch.matmul(F,F_t) # N*C*C
if normalize:
    gram = gram / (C*H*W)
return gram

# *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
```

style_loss:

```
style_loss = 0
for i,idx in enumerate(style_layers):
    style_current.append(gram_matrix(feats[idx].clone()))
    style_loss += (style_current[i] - style_targets[i]).pow(2).sum() * style_weights[i]

return style_loss

# *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
```

tv_loss:

```
def tv_loss(img, tv_weight):
    """
    Compute total variation loss.

    Inputs:
    - img: PyTorch Variable of shape (1, 3, H, W) holding an input image.
    - tv_weight: Scalar giving the weight w_t to use for the TV loss.

    Returns:
    - loss: PyTorch Variable holding a scalar giving the total variation loss
      for img weighted by tv_weight.
    """
    # Your implementation should be vectorized and not require any loops!
    # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
    N,C,H,W = img.shape
    x1 = img[:, :, 0:H-1, :]
    x2 = img[:, :, 1:H, :]
    y1 = img[:, :, :, 0:W-1]
    y2 = img[:, :, :, 1:W]
    loss = ((x2-x1).pow(2).sum() + (y2-y1).pow(2).sum()) * tv_weight
    return loss

    # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
```

结论分析与体会:

在卷积中，浅层特征更具体，深层特征则更抽象。从风格角度来说，浅层特征则记录着颜色纹理等信息，而深层特征则会记录更高级的信息。

就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟 1—3 道问答题：

在刚开始运行代码地时候，图片总是不能显示出来，会提示

```
Error: Session cannot generate requests
Error: Session cannot generate requests
cty=remote64b6d11 (c:\Users\45326\onedr
```

后面将核重新启动几次就可以了。