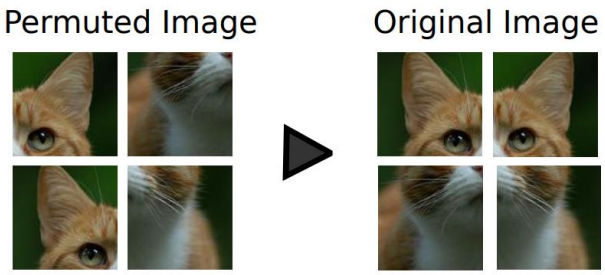
深度学习小作业 3 报告

刘翰文 522030910109

1. 图像复原任务

图像复原任务可以表述为：给定一系列被打乱的子图$x_1, x_2, \ldots, x_n$，希望从这一系列被打乱的图片中恢复其原来的顺序$x_i, x_j, \ldots, x_k$。例如，将一张图像切成2×2或3×3的一系列子图并打乱顺序，目标是设计一个模型从被打乱的图像中还原出各子图原来所在的位置。可以用一个还原2×2的图片为例子，目标是得到右侧还原的原图。

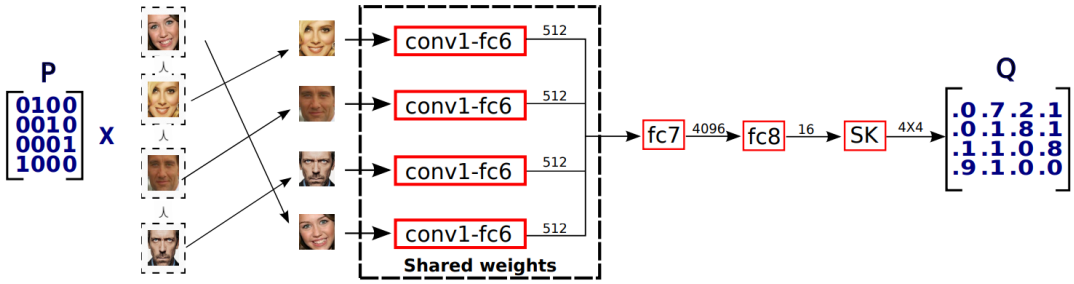


假设将一张图片分割为$N \times N$共计N^2张子图，给原图的每块子图定义$(1,2,\ldots,N^2)$的位置编号，引入排列阵$P \in \{0,1\}^{N \times N}$，其中$P(i,j) = 1$代表打乱后的第$i$张子图在原图中应排在第$j$个位置，同时$P$矩阵中每行每列只能有一个1，代表被打乱的每张子图在原图中的位置唯一。此时，图像复原任务也可以表述成设计一个模型找到一系列被打乱子图所对应的排列阵P。

2. 论文方法

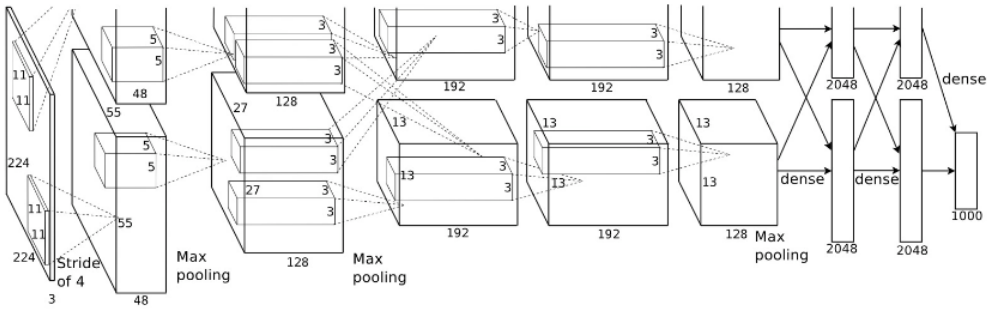
2.1 概览

在论文中，作者提出了 DeepPermNet 神经网络，它将一系列被打乱的图像序列作为输入，最终输出一个4×4的预测矩阵Q，将这个矩阵转化为符合排列阵P的格式即为网络得到的预测结果，整个网络主要包括三个部分：CNN 神经网络、全连接网络和 Sinkhorm 归一化，具体流程如下图



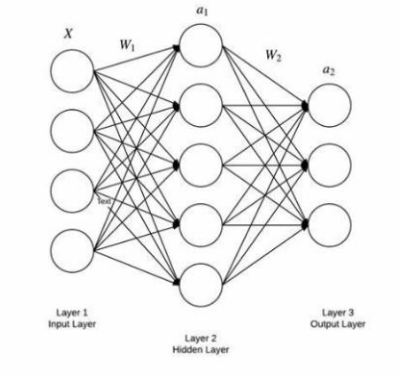
2.2 CNN 神经网络

CNN 部分主要是上图中框起来的部分，论文使用的 CNN 神经网络是 AlexNet，输入的图像序列每张并行通过 AlexNet 之中 conv1 到 fc6 的结构，包括 5 层卷积层、3 层池化层和 1 层全连接层。最终每张图像经过 AlexNet 后输出一个 512 维带有图像特征的张量。值得注意的是，每张图片通过的 AlexNet 拥有着相同的权重，以便每张图像被同等的考虑。



2.3 全连接网络

全连接网络包括了上图的 fc7 和 fc8，主要目的是将 CNN 网络得到的所有1×512图像特征的张量进行拼接，例如将上图 4 个1×512的张量拼接成一个1×2048的张量。最后通过 fc7 转为1×4096的张量，通过 fc8 转为1×16的张量，以便后续进行 Sinkhorn 算法计算预测矩阵Q。



2.4 Sinkhorn 归一化

通过全连接网络的1×16张量最终通过 Sinkhorn 归一化算法，输出一个4×4的预测矩阵Q，也称为双随机矩阵(每行每列的和固定为 1)，即$P \in [0,1]^{4\times4}$，并且满足$Q \cdot 1^4 = 1^4$以及$Q^T \times 1^4 = 1^4$。

为了描述 Sinkhorn 归一化算法，定义行归一化$R(\cdot)$和列归一化$C(\cdot)$:

$$R_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=1}^{l} Q_{i,j}};\ C_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=1}^{l} Q_{k,j}}$$

Sinkhorn 归一化算法可以表示为不断对一个矩阵进行行归一化和列归一化，使之最终能转化为一个双随机矩阵。定义损失为L，则损失对行归一化矩阵的偏导数可以表示为:

$$\frac{\partial L}{\partial Q_{p,q}} = \sum_{j=1}^{l} \frac{\partial L}{\partial R_{p,j}} \left[\frac{1_{j=q}}{\sum_{k=1}^{l} Q_{p,k}} - \frac{Q_{p,j}}{\left(\sum_{k=1}^{l} Q_{p,k}\right)^2}\right]$$

这里的Q和P分别代表行归一化的输入和输出矩阵。经过 Sinkhorn 归一化得到的双随机矩阵每行每列之和均为 1，随后需要找到和双随机矩阵最接近的排列矩阵$P \in \{0,1\}^{4\times4}$，可以将此问题描述为:

$$\hat{P} \in argmin \left|\left|\hat{P} - Q\right|\right|_F$$

$$s.t. \hat{P} \cdot 1 = 1$$

$$1^T \cdot \hat{P} = 1$$

$$\hat{P} \in \{0,1\}^{l\times l}$$

在实现中，通过取每行或每列的最大值作为 1，其他设为 0 来实现双随机矩阵到排列矩阵的转换。

3. 数据集

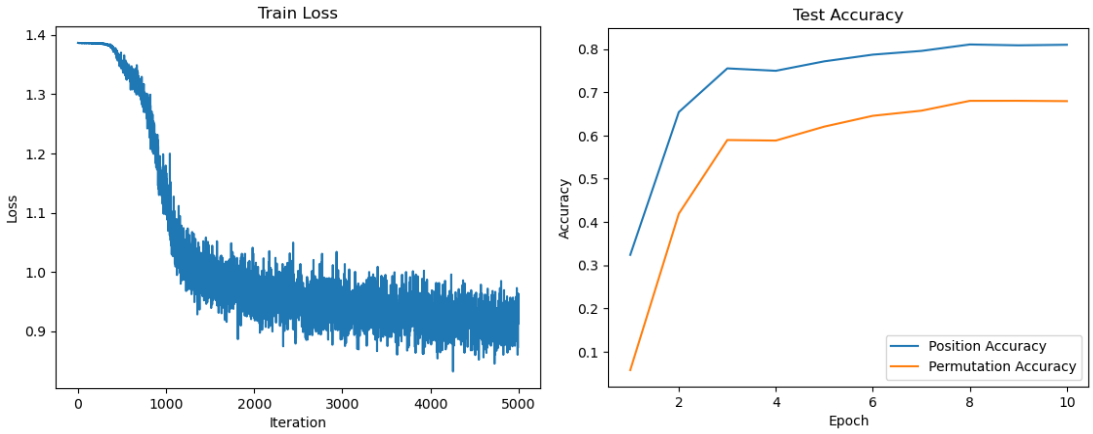
本次作业使用的数据集依旧是 CIFAR-10 数据集，唯一不同的是本次作业将每张$3 \times 32 \times 32$的图像切成 4 张$3 \times 16 \times 16$的子图并进行打乱并将这 4 张子图一起作为一个样本，并且每张图像的标签由之前的 0-9 的类别变为一个(a,b,c,d)的排列的顺序，其中a代表第一张打乱后的子图在原图中应放在a的位置。因此，数据集一共包含训练集50000个子图序列和测试机10000个子图序列。

4. 模型训练

在图像复原的任务中，选用 AlexNet 作为 CNN 网络的 DeepPermNet 作为图像复原的神经网络。选择论文中使用的 multi-class cross entropy loss 损失函数作为损失函数，选用 Adam 优化器进行优化，选择 epoch=10，学习率为 0.1 对构建的 DeepPermNet 模型进行训练，在每次 epoch 的训练后对模型进行测试，同时记录每次训练过程中的损失下降结果和测试准确率结果，测试指标主要包括两种：每张子图的还原位置正确的概率和一张图像所有子图均还原位置正确的概率。

5. 结果分析

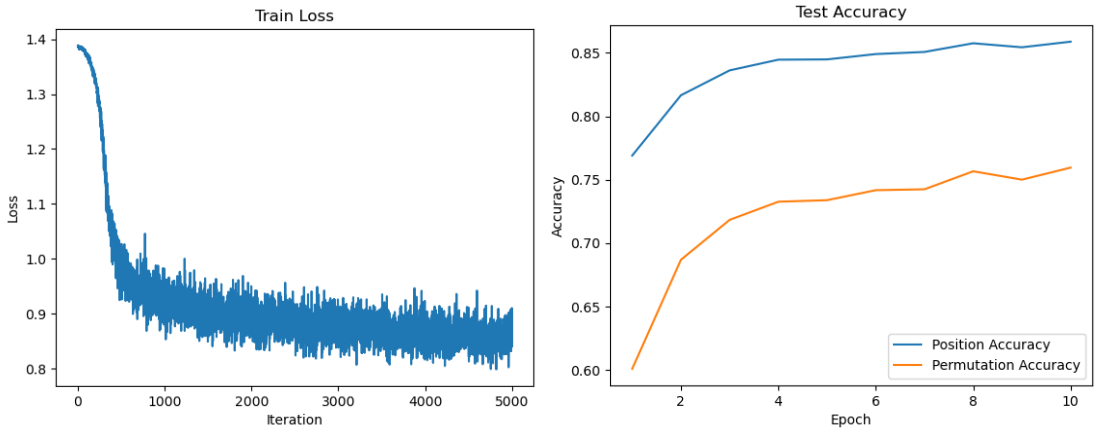
训练过程中生成的损失变化图和测试准确率结果图如下



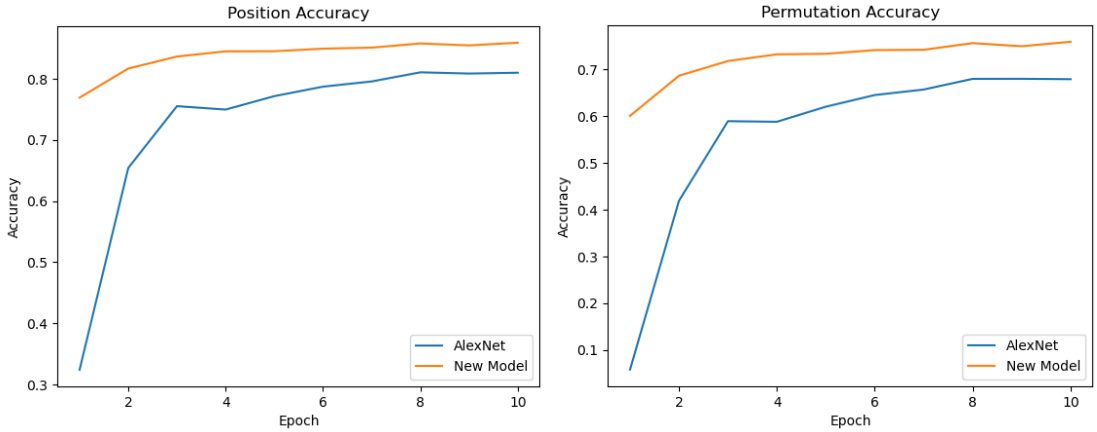
从结果的两张图看，训练的损失随迭代次数的增加下降并且逐渐趋于稳定，在测试集上的表现随 epoch 的增加也逐渐变好，并且在 10 个 epoch 之后子图和序列的准确率分别达到了 0.809 和 0.679 并逐渐趋于稳定。

6. 模型改进

论文使用的 CNN 卷积神经网络为 AlexNet，考虑到 AlexNet 的提出时间久远，性能不如现代主流的 CNN 神经网络 VGG、ResNet 等。本次作业将论文使用的 AlexNet 替换为参考 VGG 网络自己搭建的神经网络，并且不改变其他参数，对模型进行训练并且记录训练过程中的损失和测试的准确率，结果如下



分析训练损失的变化，新模型的损失在相同迭代次数下略低于原模型的损失。对于测试集上的表现，将两个模型在测试集上的准确率进行比较，得到比较结果图如下



从结果图可知，不管是所有子图复原位置的准确率，还是子图序列全部复原的准确率，用 VGG 模型改进的新模型在测试集上的表现都优于论文基于使用 AlexNet 作为 CNN 神经网络的 DeepPermNet 在测试集上的表现。在 Epoch=10 上，二者在子图复原的准确率上分别为 0.809 和 0.859，在序列复原的准确率上分别为 0.679 和 0.760。改进后的模型相较于改进前的模型在两个指标准确率上分别有了 0.05 和 0.081 的提升。

7. 总结

本次作业构建并使用了论文提出的 DeepPermNet 神经网络对子图序列进行图像复原，构建的神经网络包括 CNN 卷积神经网络、全连接网络和 Sinkhorn 归一化部分。使用子图复原和序列复原准确率作为指标对构建的模型进行了训练和测试。之后提出了对论文使用的 CNN 卷积网络部分的改进，即将 AlexNet 改为基于 VGG 的神经网络，并在测试之后证明了新模型相较于论文模型有了一定的提升。