MAE阅读小结

文章标题：$Masked\ Autoencoders\ Are\ Scable\ Vision\ Learners$，带编码的自编码器是一个可拓展的视觉学习器

写在前面：

补充一下ViT的相关背景知识：

- ViT将Transformer应用到CV领域
  - 其将图片分成多个16*16的小方块（patch）
  - 每一个方块（patch）做成一个词token放进Transformer中进行训练
- ViT证明在训练数据足够大的时候（有1000万或者1亿样本的时候）Transformer的架构精度优于CNN架构（这个在DINO中也提及了）

MAE在理解上可以理解为BERT的一个CV版本：

- MAE基于ViT
- 其将整个训练拓展到没有标号的数据上面
- 通过“完形填空”的方式来获取对图片的理解

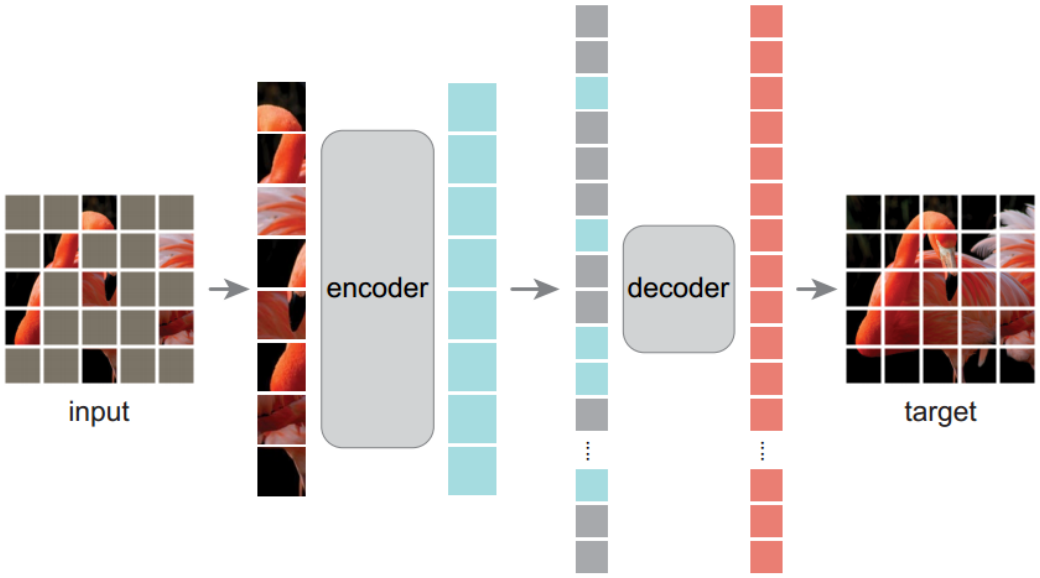
Introduction：

在引言中作者提出了重要观点：**语言和图片的信息密度是不同的。**

- 一句话去掉一些词语还能保持含义不变，这是一件较为困难的事情。这也就意味着BERT中mask的比例不能太高（“完形填空”不能挖太多空格）。
- 但是在图像中，相邻像素包含的信息常常是相似的，这意味着图像包含了较多的冗余信息。在图像中去掉一个块，往往可以通过相邻像素值进行插值还原。**MAE为了降低图片信息的冗余性，随机大比例的mask掉图片中的块**，这使得模型关注图片的全局信息而不仅仅是局部信息。

MAE的想法：**随机遮住大量的块，然后去重构这些被遮住的像素信息**，使用非对称的编码器和解码器机制。

MAE模型：



MAE的模型架构如上图所示，是一个编码器-解码器的架构。

编码器：

- 采用的是没有改动的ViT（Vision Transformer）结构，为了节省计算开销，对于**编码器来言其只看可见块**（这一点和ViT不太一样，ViT对mask块同样编码）
- 可见块token的生成方式：将可见块拿出来做线性投影+位置信息-->token

解码器：

- 解码器实际上是另外一个Transformer，由于需要重构mask块的像素信息，所以解码器需要看到可见块和mask块。可见块的编码已经通过编码器得到了，**mask块的编码可以通过一个共享的可学习到的向量来表示。**
- 论文中设计的解码器的架构是较小的，计算开销不到编码器的$\frac{1}{10}$。
- 关于解码器什么时候用的问题，解码器在预训练的时候使用或者在需要重构像素信息的时候使用。如果下游任务只想得到图片编码，那在使用的时候就不mask图像块直接用编码器进行编码就好
- 关于解码器如何重构出原始像素：
    - 解码器的最后一层是一个linear projection（这个还需了解一下，目前理解为一个线性层）
    - 如果一个patch是16*16像素，那么线性投影层会投影到256维度，接着reshape(16,16)就还原了原始像素信息
    - 损失函数采用均方误差MSELoss，**MSELoss只作用于非可见块**，对于可见块来讲图片编码器相当于看到答案了，所以MSELoss不作用与可见块

MAE的简单实现：

论文中给出了MAE简单实现的过程：

1. 对每一个输入的patch生成token：token由patch线性投影+位置信息生成

2. 随机采样：采用随机打乱操作（shuffle）操作，取打乱后的序列的前k%，论文中是25%，这意味着有75%的patch被mask掉
3. 对可见块进行编码操作。在对可见块编码完成后在其后面附加（append）和以前长度一样的mask tokens（这里就是说把被mask掉的那一部分没有被编码器编码的patch长度补回来）
4. 重新unshuffle到原来位置，3.4两点对应了MAE模型图的encoder-decoder中间部分的过程
5. 解码器解码，计算重构出来的patch和原始patch的均方误差（MSELoss），根据计算出的均方误差更新网络参数（应该是这样）

实验结论（部分）：

- MAE对数据增强不敏感
- 利用随机采样的方式采样被mask掉的块（或者说可见块，两者含义相同）效果最好
- 当被mask掉的块比例超过40%后精度会大幅提升
- 编码器不加入被mask掉的块得到的模型精度会更高同时计算量会更少
- 同时编码器和解码器采用不对称的架构模型的精度更高性能更好

初次阅读：2023.11.13