DINO阅读小结

文章标题：Emerging Properties in Self-Supervised Vision Transformers

DINO名称由来：$self-distillation\ with\ no\ labels$，self-**di**stillation with **no** labels，无标签的自蒸馏（框架）

文章理解：

Transformer作为卷积神经网络的替代品出现，由Transformer产生的Vision Transformer（ViT）具有对卷积神经网络的竞争力，但是并没有明显优势，具体表现在：需要更多的计算资源，更多的训练数据以及ViT训练出的特征没有表现出独特的特性。

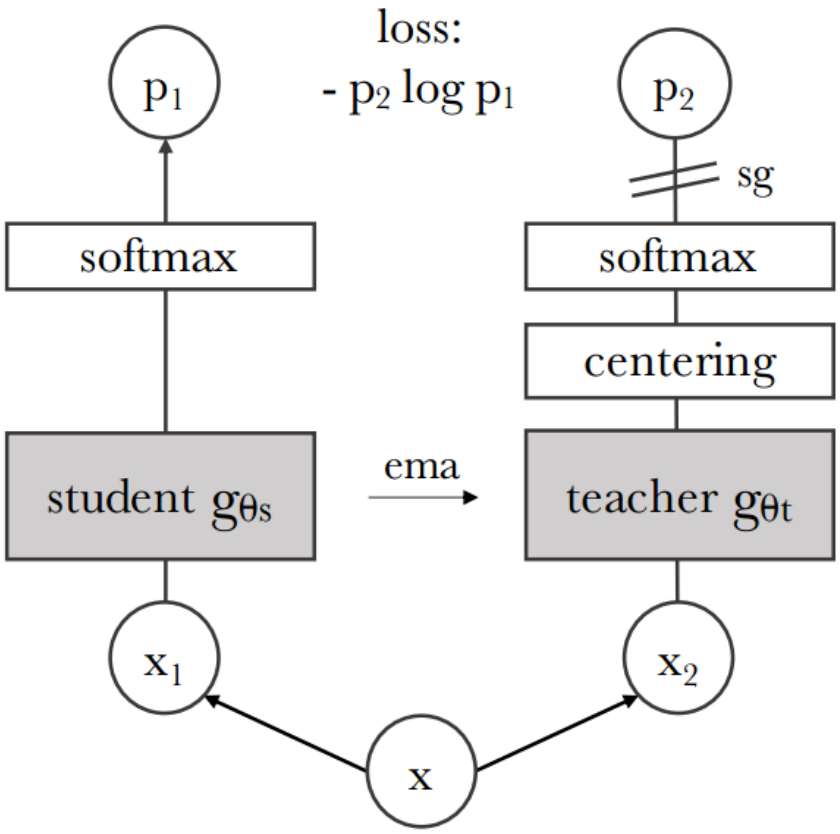
作者质疑Transformer在视觉上的成功能否用预训练中的监督来解释。其认为**Transformer在NLP中取得成功的主要因素在于使用了自监督预训练。**作者以图像中的监督学习为例：图像级的监督会将图像中包含的丰富视觉信息减少为预定义的几千类对象中的其中之一，简而言之的话就是**监督学习会使图像的视觉信息减少。**

同时作者在研究自监督的ViT时发现了几个独属于自监督ViT的属性，这些属性在监督ViT和卷积神经网络中都没有：

- 自监督ViT特征**明确包含了**场景布局，特别是**对象边界。**作者也指出这种分割掩码的出现可能是自监督方法共享的属性
- 自监督ViT在k-NN分类器上表现特别出色，不需要微调。但是在k-NN上的良好性能只有在结合某些组件的时候才会出现，如：动量编码器和多裁剪增强。

框架结构和算法描述：

基于上述发现（应该是），作者提出了DINO模型：



Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

框架的模型和算法如上两张图所示，下面结合结构和算法阐述运行过程：

1. 框架的输入是图像x，给定图像x会生成一组其不同的视图V，V中包含了两个全局视图x_1^g和x_2^g，和几个较小分辨率的局部视图，其中：**所有裁剪都通过学生网络（从损失函数公式上看这里的所有裁剪指的是局部视图，这里要确认一下），只有全局视图通过教师网络。这样设计的目的是鼓励从“局部到全局”的对应。**全局视图指的是覆盖原始图像超过50%的块，例如224* 224分辨率的块；局部视图指的是覆盖原始图像小于50%的块，例如96* 96的块。
2. 在对生成的图像生成完不同的视图后按照1中的规则放入学生网络和教师网络。
    - 学生网络和教师网络的结构完全相同，在训练过程中训练学生网络的参数$g\theta_s$以匹配教师网络参数$g\theta_t$的输出
    - 学生网络和教师网络由主干网络f和多层感知机h构成$g = f \circ h$。其中主干网络f可以是ViT[16]或者ResNet[27]，这也是作者为什么说框架具有较高灵活性的原因。**在网络训练完成后，下游任务使用的是主干网络f的输出。**
    - 从伪代码中可以看出，训练开始时，学生网络和教师网络使用的是相同的参数。图像在分别经过学生网络和教师网络后使用包含温度参数的Softmax函数进行归一化输出。**温度参数t是用来控制输出锐度(sharpness)的。**

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K}\exp(g_{\theta_s}(x)^{(k)}/\tau_s)}, \tag{1}$$

    - 为了使得学生网络的输出匹配教师网络的输出，使用交叉熵损失来学习匹配这些分布：

$$\min_{\theta_s} H(P_t(x), P_s(x)), \tag{2}$$

      H表示交叉熵损失函数
    - 为了实现1中提到的“从局部到整体的对应”，设计目标函数为，其实就是组合学生网络和教师网络学到的特征使它们交叉熵损失的和最小：

$$\min_{\theta_s} \sum_{x\in\{x_1^g, x_2^g\}} \sum_{\substack{x'\in V \\ x'\neq x}} H(P_t(x), P_s(x')). \tag{3}$$

      对该目标函数求导，采用优化方法**更新学生网络的参数。**
    - 值得注意的是，由于DINO并没有给定先验教师网络参数g_θ^t，这意味着在训练过程中也需要更新教师网络的参数，**更新教师网络的参数利用学生网络过去的迭代**（这儿翻译的不是很好，看公式就能理解了）：

$$\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$$

      其中λ遵循0.996到1的余弦schedule。
    - 其实在这里存在一个问题：没有先验的教师网络参数，学生网络学习的效果会好吗？换言之就是按照这样的方式教师网络的效果会一直比学生网络效果好，引导学生网络学习吗？论文在实验观察后得出结论，**按照上述方式更新参数，教师网络早整个训练过程中的表现由于学生，因此可以提供更高质量的目标特征指导学生网络训练**

3. 到这里，框架模型图内只有centering这一块没有解释。
    - centering是避免DINO在训练过程中崩溃的。DINO通过对教师网络的输出进行centering（居中）和sharpening（锐化，这里的锐化应该就是Softmax函数中的温度参数）操作来避免模型崩溃。
    - 居中防止一个维度占主导地位，但鼓励崩溃到均匀分布，锐化的效果与居中相反。
    - 居中可以理解为在教师网络的输出中增加了一个偏置项c：

      $$g_t(x) \leftarrow g_t(x) + c.$$

      其中c满足：

      $$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i), \qquad (4)$$

      $m > 0$是速率参数，B是批量大小。

DINO消融实验结论：

- 没有动量的情况下，框架不起作用
- 随着动量的增加，使用SK的效果影响很小
- 动量编码器对性能有重要影响
- DINO中多裁剪训练和交叉熵损失是获得良好特征的重要组成部分
- 向学生网络中添加预测器几乎没有影响
- 随着patch的减小，性能会大大提高，使用小patch获得的性能提升是以减小吞吐量为代价的

初读于2023.11.12