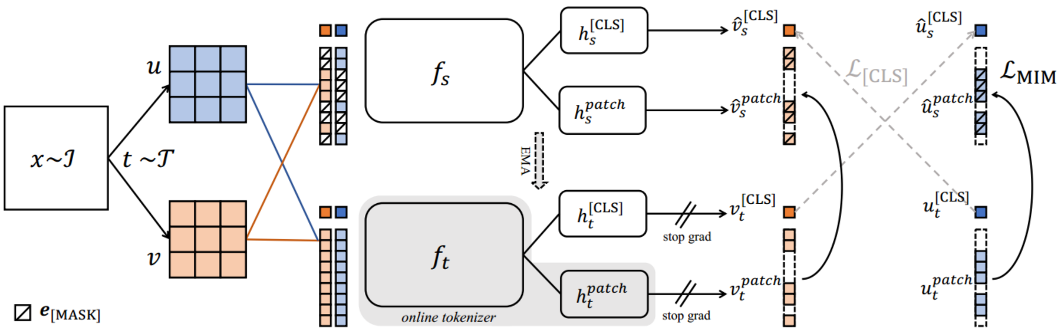
iBOT论文阅读

文章标题：$iBOT: Image\ Bert\ pre-training\ with\ Online\ Tokenizer$，可以理解为使用在线tokenizer进行图像BERT式预训练

写在前面：

iBOT可以视为BEIT和DINO的结合，在IBOT中学生网络和教师网络的输出会有两部分，一部分是和DINO中类似的$h^{[CLS]}$一部分是和BEIT中类似的h^{patch}

iBOT的模型结构图和算法流程：



Algorithm 1: iBOT PyTorch-like Pseudocode w/o multi-crop augmentation

Input:

g_s, g_t ; // student and teacher network

C, C' ; // center on [CLS] token and patch tokens

τ_s, τ_t ; // temperature on [CLS] token for student and teacher network

τ'_s, τ'_t ; // temperature on patch tokens for student and teacher network

l ; // momentum rate for network

m, m' ; // momentum rates for center on [CLS] token and patch tokens

g_t.params = g_s.params

for $\boldsymbol{x}$ in loader **do**

$\boldsymbol{u}, \boldsymbol{v}$ = augment($\boldsymbol{x}$), augment($\boldsymbol{x}$) ; // random views

$\hat{\boldsymbol{u}}, \boldsymbol{m}_u$ = blockwise_mask($\boldsymbol{u}$) ; // random block-wise masking

$\hat{\boldsymbol{v}}, \boldsymbol{m}_v$ = blockwise_mask($\boldsymbol{v}$) ; // random block-wise masking

$\hat{\boldsymbol{u}}_s^{[\texttt{CLS}]}, \hat{\boldsymbol{u}}_s^{\text{patch}} = g_s(\hat{\boldsymbol{u}}$, return_all_tok=true) ; // $[n, K]$, $[n, S^2, K]$

$\hat{\boldsymbol{v}}_s^{[\texttt{CLS}]}, \hat{\boldsymbol{v}}_s^{\text{patch}} = g_s(\hat{\boldsymbol{v}}$, return_all_tok=true) ; // $[n, K]$, $[n, S^2, K]$

$\boldsymbol{u}_t^{[\texttt{CLS}]}, \boldsymbol{u}_t^{\text{patch}} = g_t(\boldsymbol{u}$, return_all_tok=true) ; // $[n, K]$, $[n, S^2, K]$

$\boldsymbol{v}_t^{[\texttt{CLS}]}, \boldsymbol{v}_t^{\text{patch}} = g_t(\boldsymbol{v}$, return_all_tok=true) ; // $[n, K]$, $[n, S^2, K]$

$\mathcal{L}_{[\texttt{CLS}]} = \text{H}(\hat{\boldsymbol{u}}_s^{[\texttt{CLS}]}, v_t^{[\texttt{CLS}]}, C, \tau_s, \tau_t) / 2 + \text{H}(\hat{\boldsymbol{v}}_s^{[\texttt{CLS}]}, \boldsymbol{u}_t^{[\texttt{CLS}]}, C, \tau_s, \tau_t) / 2$

$\mathcal{L}_{\text{MIM}} = (\boldsymbol{m}_u \cdot \text{H}(\hat{\boldsymbol{u}}_s^{\text{patch}}, \boldsymbol{u}_t^{\text{patch}}, C', \tau'_s, \tau'_t)$.sum(dim=1) / $\boldsymbol{m}_u$.sum(dim=1) / 2

$\quad + (\boldsymbol{m}_v \cdot \text{H}(\hat{\boldsymbol{v}}_s^{\text{patch}}, \boldsymbol{v}_t^{\text{patch}}, C', \tau'_s, \tau'_t)$.sum(dim=1) / $\boldsymbol{m}_v$.sum(dim=1) / 2

($\mathcal{L}_{[\texttt{CLS}]}$.mean() + $\mathcal{L}_{\text{MIM}}$.mean()).backward()

update(g_s) ; // student, teacher and center update

g_t.params = $l \cdot g_t$.params $+ (1 - l) \cdot g_s$.params

$C = m \cdot C + (1 - m) \cdot$ cat([$\boldsymbol{u}_t^{[\texttt{CLS}]}$, $\boldsymbol{v}_t^{[\texttt{CLS}]}$]).mean(dim=0)

$C' = m' \cdot C' + (1 - m') \cdot$ cat([$\boldsymbol{u}_t^{\text{patch}}$, $\boldsymbol{v}_t^{\text{patch}}$]).mean(dim=(0, 1))

end

def H (s, t, c, τ_s, τ_t) **:**

$t = t$.detach(); // stop gradient

s = softmax(s / τ_s, dim=1)

t = softmax(($t - c$) / τ_t, dim=1); // center + sharpen

return $-(t \cdot \log(s))$.sum(dim=-1);

算法流程:

1. 与DINO类似，一开始学生网络和教师网络的参数是一致的
2. 与DINO类似，每个图像x，会经过两种不同的增广方式得到u和v
3. 与DINO不同的是，进入学生网络的图像$\hat{u}$和$\hat{v}$是u和v被mask掉的一部分后的图像
4. **在iBOT中学生网络和教师网络的输出会有两部分**:
   - 第一部分与DINO类似（暂时理解为图像的全局特征，对图像整体的编码），学生网络输出$\hat{u}_s^{[CLS]}$和$\hat{v}_s^{[CLS]}$

     教师网络输出$u_t^{[CLS]}$和$v_t^{[CLS]}$
   - 第二部分与BEIT类似（被mask掉的patch特征），学生网络输出$\hat{u}_s^{patch}$和$\hat{v}_s^{patch}$

     教师网络输出u_t^{patch}和v_t^{patch}

     第一部分的损失函数:

$$\mathcal{L}_{[\texttt{CLS}]} = \text{H}(\hat{\boldsymbol{u}}_s^{[\texttt{CLS}]}, v_t^{[\texttt{CLS}]}, C, \tau_s, \tau_t) / 2 + \text{H}(\hat{\boldsymbol{v}}_s^{[\texttt{CLS}]}, \boldsymbol{u}_t^{[\texttt{CLS}]}, C, \tau_s, \tau_t) / 2$$

和DINO文章中的损失函数是一致的：

```
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
```

H表示交叉熵损失函数，更准确的说是带有温度参数（锐度）和偏置量（居中)的交叉熵损失函数。位置和居中的平衡避免了模型在训练过程中崩溃。

第二部分的损失函数：

$$\begin{aligned}\mathcal{L}_{\text{MIM}} = & \left(\boldsymbol{m}_u \cdot \mathrm{H}(\hat{\boldsymbol{u}}_s^{\text{patch}}, \boldsymbol{u}_t^{\text{patch}}, C', \tau_s', \tau_t').\text{sum(dim=1)} \;/\; \boldsymbol{m}_u.\text{sum(dim=1)} \;/\; 2\right.\\ & + \left(\boldsymbol{m}_v \cdot \mathrm{H}(\hat{\boldsymbol{v}}_s^{\text{patch}}, \boldsymbol{v}_t^{\text{patch}}, C', \tau_s', \tau_t').\text{sum(dim=1)} \;/\; \boldsymbol{m}_v.\text{sum(dim=1)} \;/\; 2\right.\end{aligned}$$

计算被mask掉patch的特征相似度（m_u和m_v的参数含义可能是被mask掉的比例？这里需要再了解一下）

总体损失函数：

$L_{[CLS]}.mean() + L_{MIM}.mean()$

5. 参数更新：
   - 学生网络的参数更新是根据损失函数反向传播结果优化的
   - 教师网络参数更新与DINO类似，采用动量法，结合自己本来的参数以及学生网络的参数
   - 同样的，偏置量C和$C^{'}$也是采用动量法更新

   采用动量法更新的公式如下图所示：

$$\begin{aligned}& g_t.\text{params} = l \cdot g_t.\text{params} + (1-l) \cdot g_s.\text{params}\\ & C = m \cdot C + (1-m) \cdot \text{cat}([\boldsymbol{u}_t^{\texttt{[CLS]}}, \boldsymbol{v}_t^{\texttt{[CLS]}}]).\text{mean(dim=0)}\\ & C' = m' \cdot C' + (1-m') \cdot \text{cat}([\boldsymbol{u}_t^{\text{patch}}, \boldsymbol{v}_t^{\text{patch}}]).\text{mean(dim=(0, 1))}\end{aligned}$$

初读于2023.11.14