《Towards Efficient and Scalable Sharpness-Aware Minimization》CVPR2022文章阅读

[CVPR2022]https://arxiv.org/pdf/2203.02714.pdf

Improve the efficiency of SAM(propose a novel algorithm LookSAM - that only periodically calculates the inner gradient ascent, to significantly reduce the additional training cost of SAM) and apply it to large-scale training problems.

1. 背景介绍

SAM需要两个sequential (不可并行)的梯度计算，这是SAM还未用于large-batch training的原因之一。作者们提出了LookSAM以提升SAM的computational efficiency。

motivation：large-batch training的主要挑战就是倾向于收敛到sharp local minima，SAM可以有效改善这个问题

Naive idea：periodically计算第一次梯度，但是发现这种方法会导致significantly degraded performance。

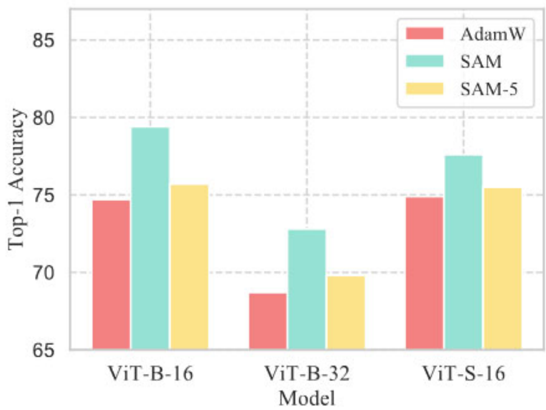

Figure 1. Accuracy of SAM-5, SAM and vanilla ViT on ImageNet-1k. SAM-5 indicates the method that calculating SAM gradients every 5 steps.

从图中可以看到，相较于SAM，SAM-5的performance显著下降。

2. LookSAM

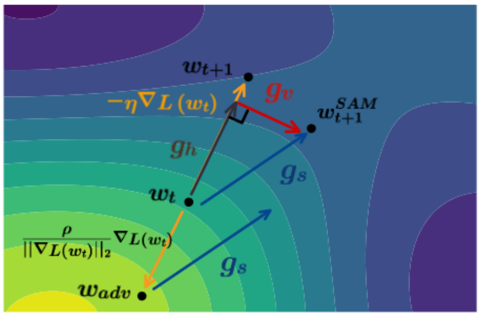

Figure 3. Visualization of LookSAM. The blue arrow g_s is SAM's gradient targeting to a flatter region. The yellow arrow $-\eta\nabla_{w}\mathcal{L}_S(w)$ indicates the SGD gradient. g_h (the brown arrow) and g_v (the red arrow) are the orthogonal gradient components of g_s, parallel and vertical to the SGD gradient, respectively.

SAM的gradient:

$$g_s = \nabla_w \mathcal{L}_S(w)|_{w+\hat{\epsilon}}$$

对其进行taylor展开得到:

$$\begin{aligned}
\nabla_{\boldsymbol{w}}\mathcal{L}_S(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\epsilon}} &= \nabla_{\boldsymbol{w}}\mathcal{L}_S(\boldsymbol{w}+\hat{\epsilon}) \\
&\approx \nabla_{\boldsymbol{w}}[\mathcal{L}_S(\boldsymbol{w}) + \hat{\epsilon}\cdot\nabla_{\boldsymbol{w}}\mathcal{L}_S(\boldsymbol{w})] \\
&= \nabla_{\boldsymbol{w}}[\mathcal{L}_S(\boldsymbol{w}) + \frac{\rho}{\|\nabla_{\boldsymbol{w}}\mathcal{L}_S(\boldsymbol{w})\|}\nabla_{\boldsymbol{w}}\mathcal{L}_S(\boldsymbol{w})\cdot\nabla_{\boldsymbol{w}}\mathcal{L}_S(\boldsymbol{w})^T] \\
&= \nabla_{\boldsymbol{w}}[\mathcal{L}_S(\boldsymbol{w}) + \rho\,\|\nabla_{\boldsymbol{w}}\mathcal{L}_S(\boldsymbol{w})\|]
\end{aligned}$$

SAM的梯度由两部分组成，$\nabla_w\mathcal{L}_S(w)$和$||\nabla_w\mathcal{L}_S(w)||$的梯度，作者们认为优化梯度的 L2-范数可以提示模型收敛到平坦区域，因为平坦区域通常意味着低梯度范数值。SAM的更新可以分为两个部分：第一部分用于降低loss(g_h)，第二部分用于导向更平坦的区域(g_v)。g_h是普通的 SGD 的梯度方向，即使没有SAM，也需要在每一步计算。因此，SAM的额外计算代价主要是由第二部分g_v引起的。已知SAM的梯度(蓝色箭头)和SGD的梯度方向g_h，我们可以进行投影得到g_v:

$$g_v = \nabla_w\mathcal{L}_S(w)|_{w+\hat{\epsilon}}\cdot\sin(\theta)$$

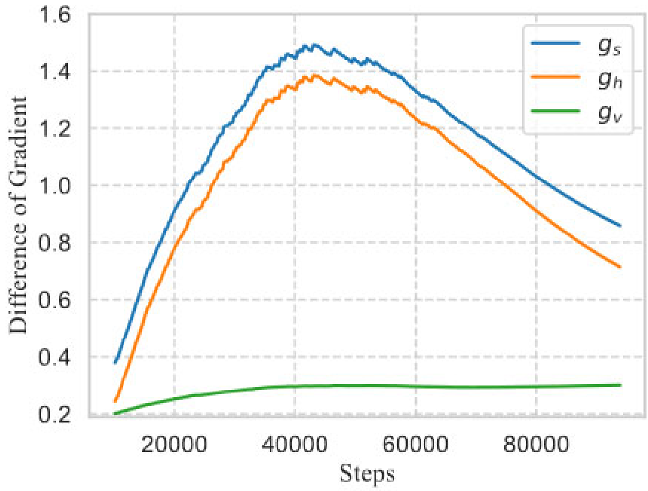

Figure 2. Difference of gradients between every 5 steps for $\boldsymbol{g_s}$, $\boldsymbol{g_h}$, and $\boldsymbol{g_v}$ (i.e., $||g_s^t - g_s^{t+k}||$). $\boldsymbol{g_v}$ that leads to a smoother region changes much slower than $\boldsymbol{g_s}$ and $\boldsymbol{g_h}$.

观察得到，g_v的变化比g_h和g_s慢得多，因此作者们对g_v的计算次数进行了优化

Algorithm 1 LookSAM

Input: $x \in \mathbb{R}^d$, learning rate η_t, update frequency k.
for $t \leftarrow 1$ **to** T **do**
  Sample Minibatch $\mathcal{B} = \{(x_i, y_i), \cdots, (x_{|\mathcal{B}|}, y_{|\mathcal{B}|})\}$ from X.
  Compute gradient $\boldsymbol{g} = \nabla_{\boldsymbol{w}} \mathcal{L}_{\mathcal{B}}(\boldsymbol{w})$ on minibatch $\mathcal{B}$.
  if t%k = 0 **then**
    Compute $\epsilon(\boldsymbol{w}) = \rho \cdot \nabla_{\boldsymbol{w}} \mathcal{L}_S(\boldsymbol{w}) / \|\nabla_{\boldsymbol{w}} \mathcal{L}_S(\boldsymbol{w})\|$
    Compute SAM gradient: $\boldsymbol{g_s} = \nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})|_{\boldsymbol{w}+\epsilon(\boldsymbol{w})}$
    $\boldsymbol{g_v} = \boldsymbol{g_s} - \|\boldsymbol{g_s}\| \cos(\theta) \cdot \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}$, where $\cos(\theta) = \frac{\boldsymbol{g} \cdot \boldsymbol{g_s}}{\|\boldsymbol{g}\| \|\boldsymbol{g_s}\|}$
  else
    $\boldsymbol{g_s} = \boldsymbol{g} + \alpha \cdot \frac{\|\boldsymbol{g}\|}{\|\boldsymbol{g_v}\|} \cdot \boldsymbol{g_v}$
  end if
  Update weights: $\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \eta_t \cdot \boldsymbol{g_s}$
end for

3. LAYER-WISE LOOKSAM

在 SAM 的内部最大化中引入layer-wise scaling

Algorithm 2 Look-LayerSAM

Input: $x \in \mathbb{R}^d$, learning rate η_t, update frequency k.
for $t \leftarrow 1$ **to** T **do**
  Sample Minibatch $\mathcal{B} = \{(x_i, y_i), \cdots, (x_{|\mathcal{B}|}, y_{|\mathcal{B}|})\}$ from X.
  Compute gradient $\boldsymbol{g} = \nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})$ on minibatch $\mathcal{B}$.
  if t%k = 0 **then**
    Compute $\epsilon(\boldsymbol{w})^{(i)} = \rho \frac{\|\boldsymbol{w}^{(i)}\|}{\|\boldsymbol{g}^{(i)}\|} \cdot \text{sign}(\boldsymbol{g})$
    Compute SAM gradient: $\boldsymbol{g_s} = \nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})|_{\boldsymbol{w}+\epsilon(\boldsymbol{w})}$
    $\boldsymbol{g_v} = \boldsymbol{g_s} - \|\boldsymbol{g_s}\| \cos(\theta) \cdot \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}$, where $\cos(\theta) = \frac{\boldsymbol{g} \cdot \boldsymbol{g_s}}{\|\boldsymbol{g}\| \|\boldsymbol{g_s}\|}$
  else
    $\boldsymbol{g_s} = \boldsymbol{g} + \alpha \cdot \frac{\|\boldsymbol{g}\|}{\|\boldsymbol{g_v}\|} \cdot \boldsymbol{g_v}$
  end if
  Update weights: $\boldsymbol{w}_{\boldsymbol{t+1}}^{(i)} = \boldsymbol{w}_{\boldsymbol{t}}^{(i)} - \eta_t^{(i)} \cdot \boldsymbol{g}_{\boldsymbol{s}}^{(i)}$
end for