《Make Sharpness-Aware Minimization Stronger: A Sparsified Perturbation Approach》-NIPS2022

[NIPS2022]https://arxiv.org/pdf/2210.05177.pdf

Improved SAM algorithm based on sparse perturbation

1. 背景介绍

SAM的方法有一个缺点，那就是它需要对所有参数进行计算和扰动，这导致了大量的计算开销。为了解决这个问题，论文提出了稀疏SAM（SSAM）的方法。SSAM的核心思想是通过一个二进制掩码，只对部分参数进行扰动和计算，从而实现稀疏扰动。

为了得到这个稀疏掩码，论文给出了两种方法，一种是基于Fisher信息的方法，另一种是基于动态稀疏训练的方法。论文还从理论上证明了，即使只对部分参数进行扰动和计算，SSAM也能像原始的SAM一样，以$O\left(\log T / \sqrt{T}\right)$的速率收敛。

通过这种方法，SSAM不仅可以大大减少计算开销，还能有效地提高模型的泛化能力。实验结果也证实了这一点，表明在CIFAR10、CIFAR100和ImageNet-1K上，SSAM比SAM更高效，性能在只有50%稀疏度的扰动下被保留或者甚至更好。

Fisher Information: measure the information that an observable random variable carries about an unknown parameter of a distribution; 作者使用它来衡量一个weight是否需要进行扰动以获得更flat的minima。

Dynamic sparse training: pruning unimportant weights。

Rethinking the Perturbation in SAM: SAM perturbs all parameters indiscriminately, but the fact is that merely about 5% parameter space is sharp while the rest is flat。作者发现，有些维度不进行perturbation反而能获得更好的generalization。并且作者还对比了SAM和SGD，发现两者的大部分梯度没有显著差异。由此，作者认为most parameters of the model require no perturbations for achieving the flat minima。

2. Methodology

2.1. SSAM

稀疏SAM (SSAM) 使用一个稀疏二进制掩码来决定哪些参数应该被扰动，也就是将ϵ乘稀疏的二进制掩码m，目标函数可被改写成：

$$\min_{\boldsymbol{w}} f_{\mathcal{S}}\left(\boldsymbol{w}+\rho \cdot \frac{\nabla_{\boldsymbol{w}} f(\boldsymbol{w})}{||\nabla_{\boldsymbol{w}} f(\boldsymbol{w})||_2} \odot \boldsymbol{m}\right)$$

并且m在训练过程中会被定期更新。作者提供了两种获取稀疏掩码m的解决方案，分别是基于Fisher信息的稀疏SAM (SSAM-F)和基于动态稀疏训练的稀疏SAM (SSAM-D)。

Algorithm 1 Sparse SAM (SSAM)

Input: sparse ratio s, dense model $\boldsymbol{w}$, binary mask $\boldsymbol{m}$, update interval T_m, number of samples N_F, learning rate η, training set $\mathcal{S}$.

Initialize $\boldsymbol{w}$ and $\boldsymbol{m}$ randomly.
for epoch $t = 1, 2 \dots T$ **do**
  for each training iteration **do**
    Sample a batch from $\mathcal{S}$: $\mathcal{B}$
    Compute perturbation ϵ by Eq. (2)
    if $t \mod T_m = 0$ **then**
      Generate mask m via Option I or II
    end if
    $\epsilon \leftarrow \epsilon \odot \boldsymbol{m}$
  end for
  $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla f(\boldsymbol{w} + \epsilon)$
end for
return Final weight of model $\boldsymbol{w}$

Algorithm 2 Sparse Mask Generation

Option I:(Fisher Information Mask)
Sample N_F data from $\mathcal{S}$: $\mathcal{B}_F$
Compute Fisher $\hat{F}_{\boldsymbol{w}}$ by Eq. (5)
$\boldsymbol{m_1} = \{m_i = 1 | m_i \in \boldsymbol{m}\} \leftarrow \text{ArgTopK}(\hat{F}_{\boldsymbol{w}}, s \cdot |\boldsymbol{w}|)$
$\boldsymbol{m_0} = \{m_i = 0 | m_i \in \boldsymbol{m}\} \leftarrow \{m_i | m_i \notin \boldsymbol{m_1}\}$
$\boldsymbol{m} \leftarrow \boldsymbol{m_0} \cup \boldsymbol{m_1}$
Option II:(Dynamic Sparse Mask)
$N_{drop} = f_{decay}(t; \alpha) \cdot (1 - s) \cdot |\boldsymbol{w}|$
$N_{growth} = N_{drop}$
$\boldsymbol{m_1} = \{m_i = 1 | m_i \in \boldsymbol{m}\} \leftarrow \{m_i = 1 | m_i \in \boldsymbol{m}\} - \text{ArgTopK}_{m_i \in \boldsymbol{m_1}}(-|\nabla f(\boldsymbol{w})|, N_{drop})$
$\boldsymbol{m_1} \leftarrow \{m_i = 1 | m_i \in \boldsymbol{m}\} + \text{Random}_{m_i \notin \boldsymbol{m_1}}(N_{growth})$
$\boldsymbol{m_0} = \{m_i = 0 | m_i \in \boldsymbol{m}\} \leftarrow \{m_i | m_i \notin \boldsymbol{m_1}\}$
$\boldsymbol{m} \leftarrow \boldsymbol{m_0} \cup \boldsymbol{m_1}$
return Sparse mask $\boldsymbol{m}$

2.2. SSAM-F

The Fisher information is defined by:

$$F_{\boldsymbol{w}} = \mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{y \sim p_{\boldsymbol{w}}(y|x)} \nabla_{\boldsymbol{w}} \log p_{\boldsymbol{w}}(y|x) \nabla_{\boldsymbol{w}} \log p_{\boldsymbol{w}}(y|x)^T \right]$$

因为$p(x)$不是available的，所以作者使用sampling来估计：

$$F_{\boldsymbol{w}} = \frac{1}{N_F} \mathbb{E}_{y \sim p_{\boldsymbol{w}}(y|\boldsymbol{x}_i)} (\nabla_{\boldsymbol{w}} \log p_{\boldsymbol{w}}(y|\boldsymbol{x}_i))^2$$

empirical Fisher:

$$\hat{F}_{\boldsymbol{w}} = \frac{1}{N_F} (\nabla_{\boldsymbol{w}} \log p_{\boldsymbol{w}}(y_i|\boldsymbol{x}_i))^2$$

获得经验费雪信息后按照元素降序排序，取前k个元素对应的参数添加扰动，掩码相应位置设置为1.

2.3. SSAM-D

由于费雪信息矩阵计算相对较高，本文也给出了基于动态稀疏训练的算法。动态稀疏训练包括扰动丢弃(Perturbation Dropping)步骤和扰动增长(Perturbation Growth)步骤。
扰动丢弃步骤：丢弃最平坦维度的权重。衡量指标即对应梯度的绝对值。
扰动增长步骤：核心目标是探索尽可能多的扰动组合，主要方法即随机选取一定数量的参数添加扰动。