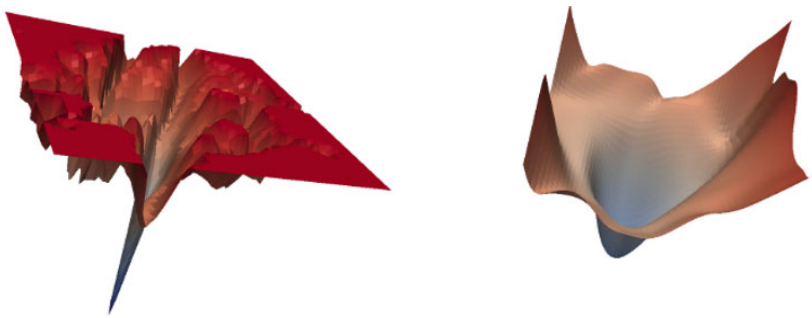
《SHARPNESS-AWARE MINIMIZATION FOR EFFICIENTLY IMPROVING GENERALIZATION》-ICLR2021文章阅读

[ICLR 21] https://arxiv.org/pdf/2010.01412.pdf sharpness aware minimization, which is formulated as a min-max optimization problem.

SAM优化器（Sharpness-Aware Minimization），最小化 loss value 时，同时最小化 loss sharpness，尝试减小在训练集S上训练到在整个数据分布D上的泛化误差。也就是说，SAM旨在找到邻域内的loss value也相对低的parameters，也就可以将问题转化为minmax问题：找到邻域内的最大的loss，然后最小化这个最大的loss。

1. 背景介绍

如今的神经网络模型大多使用了过多的参数(overparameterized)，因此模型的泛化能力无法保证，通常认为，收敛flat minima的模型相较于收敛到sharp minima的模型具有更好的泛化性能。尽管这种联系为实现更好泛化的新模型训练方法提供了可能，但在当时还没有找到在一系列最先进的模型上都能有效提高泛化能力的实用高效算法。这篇文章中提出了SAM来提高模型的泛化能力，这种方法直接利用损失图景的几何形状及其与泛化的联系。



左右两张图对比，sharp和flat。

2. SAM

使用PAC-贝叶斯泛化误差上界理论，并且在给定条件下可以得到以下不等式：

$$L_{\mathscr{D}}(w) \leqslant \max_{||\epsilon||_2 \leqslant \rho} L_S(w+\epsilon) + h\left(||w||_2^2 / \rho^2\right)$$

考虑到具体的函数h受证明细节的影响严重，使用正则化项$\lambda||w||_2^2$代替$h\left(||w||_2^2 / \rho^2\right)$，为了得到损失函数的极小解，就是求右边式子的最小值，得到**需要优化的问题如下（也就是一个极小-极大化minimax问题）**：

$$\min_w L_S^{SAM}(w) + \lambda||w||_2^2 \qquad where \qquad L_S^{SAM}(w) \triangleq \max_{||\epsilon||_p \leqslant \rho} L_S(w+\epsilon).$$

目的是使得找到的minima更flat，需要找到一个w的代表区域（找最大扰动）。为了最小化$L_S^{SAM}(w)$，作者提出可以通过对inner maximization求微分来得到$\nabla_w L_S^{SAM}(w)$，然后就可以用SGD方法来得到$\min_w L_S^{SAM}(w)$。

inner maximization: 在固定损失函数后，在训练集上更新模型超参数使得损失函数值最大，得到当前最优的模型参数。对应的数学表达式就是[超参数] = arg max [损失函数]。

该问题的inner maximization: $\epsilon^*(w) = \arg\max_{||\epsilon||_p \leqslant \rho} L_S(w+\epsilon)$. 也就是在$w$的$\epsilon$邻域$[a-\epsilon, a+\epsilon]$中，找到使得目标函数$L_S(w+\epsilon)$最大的参数$\epsilon$. 该问题可以进行进一步的化简来进行求解。

简单推导过程：

先进行一阶泰勒展开

$$\boldsymbol{\epsilon}^*(\boldsymbol{w}) \triangleq \underset{\|\boldsymbol{\epsilon}\|_p \leq \rho}{\arg\max}\, L_{\mathcal{S}}(\boldsymbol{w}+\boldsymbol{\epsilon}) \approx \underset{\|\boldsymbol{\epsilon}\|_p \leq \rho}{\arg\max}\, L_{\mathcal{S}}(\boldsymbol{w}) + \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w}) = \underset{\|\boldsymbol{\epsilon}\|_p \leq \rho}{\arg\max}\, \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w}).$$

利用对偶范数的解：$||z||_* := \sup\left\{z^T x : x \in \mathbb{R}^n, ||x|| \leqslant 1\right\}$，在该问题里，$\epsilon$相当于$x$，$\nabla_w L_S(w)$相当于$z$，从而解得:

$$\hat{\boldsymbol{\epsilon}}(\boldsymbol{w}) = \rho\, \text{sign}\left(\nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})\right) \left|\nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})\right|^{q-1} / \left(\left\|\nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})\right\|_q^q\right)^{1/p}$$

其中$1/p + 1/q = 1$.
文章中提到$p = 2$时效果通常最优，将$p = q = 2$代入，得到：

$$\begin{aligned}\hat{\epsilon}(w) &= \rho \text{sign}(\nabla_w L_S(w)) |\nabla_w L_S(w)| / ||\nabla_w L_S(w)||_2 \\ &= \rho \frac{\nabla_w L_S(w)}{||\nabla_w L_S(w)||_2}.\end{aligned}$$

从而得到一个$\nabla_w L_S^{SAM}(w)$的近似：

$$\begin{aligned}\nabla_{\boldsymbol{w}} L_{\mathcal{S}}^{SAM}(\boldsymbol{w}) &\approx \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w} + \hat{\boldsymbol{\epsilon}}(\boldsymbol{w})) = \frac{d(\boldsymbol{w} + \hat{\boldsymbol{\epsilon}}(\boldsymbol{w}))}{d\boldsymbol{w}} \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(w)} \\ &= \nabla_w L_{\mathcal{S}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})} + \frac{d\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}{d\boldsymbol{w}} \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}.\end{aligned}$$

将二阶导忽略，得到:

$$\nabla_{\boldsymbol{w}} L_{\mathcal{S}}^{SAM}(\boldsymbol{w}) \approx \nabla_{\boldsymbol{w}} L_{\mathcal{S}}(w)|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}$$

作者额外还做了加入二阶导项的实验，发现表现变差。
SAM算法的具体步骤如下：

Input: Training set $\mathcal{S} \triangleq \cup_{i=1}^{n}\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$, Loss function $l : \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, Batch size b, Step size $\eta > 0$, Neighborhood size $\rho > 0$.
Output: Model trained with SAM
Initialize weights $\boldsymbol{w}_0$, $t = 0$;
while *not converged* **do**
  Sample batch $\mathcal{B} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ...(\boldsymbol{x}_b, \boldsymbol{y}_b)\}$;
  Compute gradient $\nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})$ of the batch's training loss;
  Compute $\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})$ per equation 2;
  Compute gradient approximation for the SAM objective (equation 3): $\boldsymbol{g} = \nabla_w L_{\mathcal{B}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}$;
  Update weights: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}$;
  $t = t + 1$;
end
return $\boldsymbol{w}_t$

Algorithm 1: SAM algorithm

先基于参数w对batch data S計算gradient G。然后求解 G 的 dual norm，依照 dual vector 方向更新参数，得到$w + \epsilon$。再基于参数$w + \epsilon$对S计算gradient G'。最后用 G' 更新原本的的参数w。
文章中给出了图示：

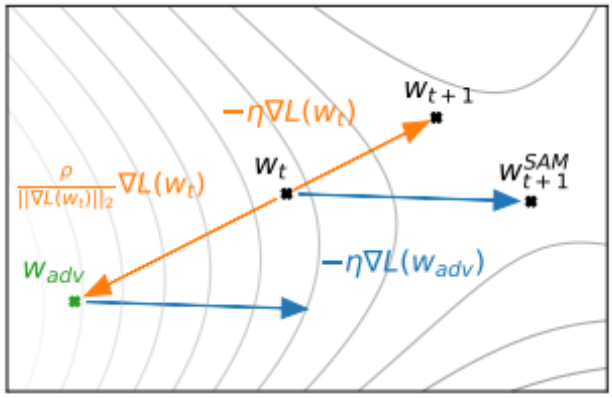


Figure 2: Schematic of the SAM parameter update.

3. 实验结果

SAM 只有一个超参数ρ，用来定义计算 sharpness 的邻域大小 (neighborhood size)。论文中实验数据的ρ是由grid search从{0.01, 0.02, 0.05, 0.1, 0.2, 0,5}中搜出来的。由于SAM的一次参数更新中包含了两次BP过程，在实验上让其他方法都训练了两倍的epoch，以保证公平。

3.1. Image Classification

作者使用不同的网络架构并搭配不同的data augmentation方法，共设计了五组实验，结果如下：

		CIFAR-10		CIFAR-100	
Model	Augmentation	SAM	SGD	SAM	SGD
WRN-28-10 (200 epochs)	Basic	**2.7**$_{\pm 0.1}$	3.5$_{\pm 0.1}$	**16.5**$_{\pm 0.2}$	18.8$_{\pm 0.2}$
WRN-28-10 (200 epochs)	Cutout	**2.3**$_{\pm 0.1}$	2.6$_{\pm 0.1}$	**14.9**$_{\pm 0.2}$	16.9$_{\pm 0.1}$
WRN-28-10 (200 epochs)	AA	**2.1**$_{\pm <0.1}$	2.3$_{\pm 0.1}$	**13.6**$_{\pm 0.2}$	15.8$_{\pm 0.2}$
WRN-28-10 (1800 epochs)	Basic	**2.4**$_{\pm 0.1}$	3.5$_{\pm 0.1}$	**16.3**$_{\pm 0.2}$	19.1$_{\pm 0.1}$
WRN-28-10 (1800 epochs)	Cutout	**2.1**$_{\pm 0.1}$	2.7$_{\pm 0.1}$	**14.0**$_{\pm 0.1}$	17.4$_{\pm 0.1}$
WRN-28-10 (1800 epochs)	AA	**1.6**$_{\pm 0.1}$	2.2$_{\pm <0.1}$	**12.8**$_{\pm 0.2}$	16.1$_{\pm 0.2}$
Shake-Shake (26 2x96d)	Basic	**2.3**$_{\pm <0.1}$	2.7$_{\pm 0.1}$	**15.1**$_{\pm 0.1}$	17.0$_{\pm 0.1}$
Shake-Shake (26 2x96d)	Cutout	**2.0**$_{\pm <0.1}$	2.3$_{\pm 0.1}$	**14.2**$_{\pm 0.2}$	15.7$_{\pm 0.2}$
Shake-Shake (26 2x96d)	AA	**1.6**$_{\pm <0.1}$	1.9$_{\pm 0.1}$	**12.8**$_{\pm 0.1}$	14.1$_{\pm 0.2}$
PyramidNet	Basic	**2.7**$_{\pm 0.1}$	4.0$_{\pm 0.1}$	**14.6**$_{\pm 0.4}$	19.7$_{\pm 0.3}$
PyramidNet	Cutout	**1.9**$_{\pm 0.1}$	2.5$_{\pm 0.1}$	**12.6**$_{\pm 0.2}$	16.4$_{\pm 0.1}$
PyramidNet	AA	**1.6**$_{\pm 0.1}$	1.9$_{\pm 0.1}$	**11.6**$_{\pm 0.1}$	14.6$_{\pm 0.1}$
PyramidNet+ShakeDrop	Basic	**2.1**$_{\pm 0.1}$	2.5$_{\pm 0.1}$	**13.3**$_{\pm 0.2}$	14.5$_{\pm 0.1}$
PyramidNet+ShakeDrop	Cutout	**1.6**$_{\pm <0.1}$	1.9$_{\pm 0.1}$	**11.3**$_{\pm 0.1}$	11.8$_{\pm 0.2}$
PyramidNet+ShakeDrop	AA	**1.4**$_{\pm <0.1}$	1.6$_{\pm <0.1}$	**10.3**$_{\pm 0.1}$	10.6$_{\pm 0.1}$

Table 1: Results for SAM on state-of-the-art models on CIFAR-{10, 100} (WRN = WideResNet; AA = AutoAugment; SGD is the standard non-SAM procedure used to train these models).

从结果来看，不管哪一种配置下，加上SAM都比只使用SGD效果更好。
之后作者进一步在ResNet上跑了ImageNet实验，同样的，加上SAM后效果更好。并且，还发现随着epoch的增加，SAM在减缓overfitting上也有明显优势。

Model	Epoch	SAM		Standard Training (No SAM)	
		Top-1	Top-5	Top-1	Top-5
ResNet-50	100	**22.5**$_{\pm 0.1}$	6.28$_{\pm 0.08}$	22.9$_{\pm 0.1}$	6.62$_{\pm 0.11}$
	200	**21.4**$_{\pm 0.1}$	5.82$_{\pm 0.03}$	22.3$_{\pm 0.1}$	6.37$_{\pm 0.04}$
	400	**20.9**$_{\pm 0.1}$	5.51$_{\pm 0.03}$	22.3$_{\pm 0.1}$	6.40$_{\pm 0.06}$
ResNet-101	100	**20.2**$_{\pm 0.1}$	5.12$_{\pm 0.03}$	21.2$_{\pm 0.1}$	5.66$_{\pm 0.05}$
	200	**19.4**$_{\pm 0.1}$	4.76$_{\pm 0.03}$	20.9$_{\pm 0.1}$	5.66$_{\pm 0.04}$
	400	**19.0**$_{\pm <0.01}$	4.65$_{\pm 0.05}$	22.3$_{\pm 0.1}$	6.41$_{\pm 0.06}$
ResNet-152	100	**19.2**$_{\pm <0.01}$	4.69$_{\pm 0.04}$	20.4$_{\pm <0.0}$	5.39$_{\pm 0.06}$
	200	**18.5**$_{\pm 0.1}$	4.37$_{\pm 0.03}$	20.3$_{\pm 0.2}$	5.39$_{\pm 0.07}$
	400	**18.4**$_{\pm <0.01}$	4.35$_{\pm 0.04}$	20.9$_{\pm <0.0}$	5.84$_{\pm 0.07}$

Table 2: Test error rates for ResNets trained on ImageNet, with and without SAM.

3.2. Finetuning

对基于ImageNet pretrain好的模型进行了finetuning的实验，发现SAM依然提升了模型的表现。

3.3. Robustness to label noise

作者还进一步探究了SAM的鲁棒性，通过将一定比例的training set中的label给错，得到的结果如下：

Method	Noise rate (%)			
	20	40	60	80
Sanchez et al. (2019)	94.0	92.8	90.3	74.1
Zhang & Sabuncu (2018)	89.7	87.6	82.7	67.9
Lee et al. (2019)	87.1	81.8	75.4	-
Chen et al. (2019)	89.7	-	-	52.3
Huang et al. (2019)	92.6	90.3	43.4	-
MentorNet (2017)	92.0	91.2	74.2	60.0
Mixup (2017)	94.0	91.5	86.8	76.9
MentorMix (2019)	**95.6**	**94.2**	91.3	**81.0**
SGD	84.8	68.8	48.2	26.2
Mixup	93.0	90.0	83.8	70.2
Bootstrap + Mixup	93.3	92.0	87.6	72.0
SAM	95.1	93.4	90.5	77.9
Bootstrap + SAM	95.4	**94.2**	**91.8**	79.9

Table 4: Test accuracy on the clean test set for models trained on CIFAR-10 with noisy labels. Lower block is our implementation, upper block gives scores from the literature, per Jiang et al. (2019).

尽管SAM并没有刷出更高的成绩，但是单纯使用SAM就可以达到接近SOTA的效果。