Sharpness-Aware Minimization Activates the Interactive Teaching's Understanding and Optimization

Mingwei Xu [1], Xiaofeng Cao [1,✉], Ivor W. Tsang [2,3]

[1] School of Artificial Intelligence, Jilin University, China
[2] CFAR and IHPC, Agency for Science, Technology and Research (A*STAR), Singapore
[3] College of Computing and Data Science, Nanyang Technological University, Singapore

Backgrounds

- Teaching recognized as a pervasive mechanism for disseminating knowledge within human society, has found extensive application in contemporary deep learning methodologies.

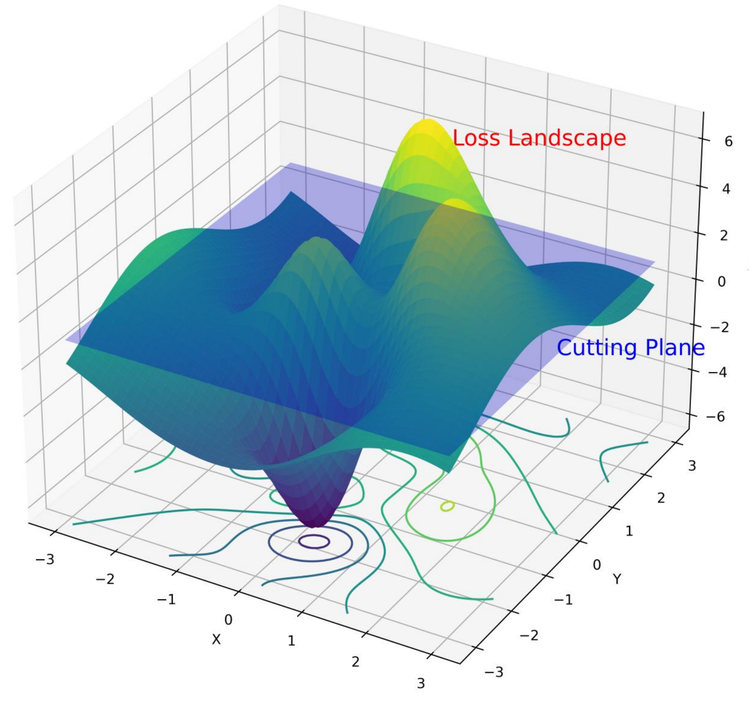
- It serves as a cornerstone for various techniques such as knowledge distillation, data distillation, model compression, and machine teaching, facilitating optimal training control.

- Recent investigations into pedagogy have illuminated the integration of large language models and multi-agent systems into educational frameworks.

Interactive teaching's understanding

- Interactive teaching methods like co-teaching update parameters by reducing high loss values in the landscape.

- By actively involving two teachers, models in interactive framework learn from each other's strengths through a collaborative filtering mechanism and focus on minimizing loss examples.



Introduction

- The underlying optimization principles and convergence of interactive teaching lack theoretical analysis, and co-teaching serves as a notable prototype in this regard.
- In this paper, we discuss its role as a reduction of the larger loss landscape derived from Sharpness-Aware Minimization (SAM).
- Then, we classify it as an iterative parameter estimation process using Expectation-Maximization. The convergence of this typical interactive teaching is achieved by continuously optimizing a variational lower bound on the log marginal likelihood.

Our optimization

- To further enhance interactive teaching's performance, we incorporate SAM's strong generalization information into interactive teaching, referred as Sharpness Reduction Interactive Teaching (SRIT). This integration can be viewed as a novel sequential optimization process.
- We validate the performance of our approach through multiple experiments.

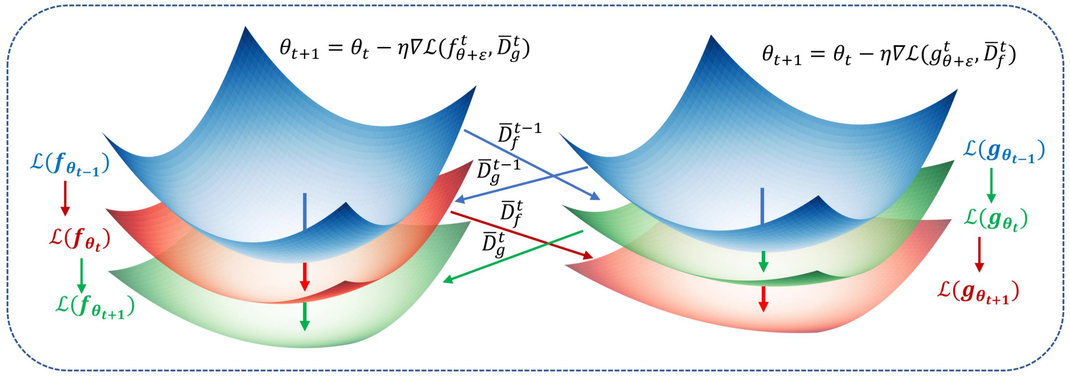


Figure1: Interactive teaching, Sharpness Reduction Interactive Teaching (SRIT), the plane in the figure represents the loss landscape, which gradually becomes flat during the iterative optimization process due to the receipt of flat gradient information cues from each other.

Theoretical Analysis

- Our core assumption is that the cleanliness of the data distribution serves as a latent variable, as it remains unknown within the training dataset.
- Based on this assumption, the interactive teaching process can be effectively exemplified as a unique type of parameter iteration within the EM framework.
- This perspective provides a probabilistic modeling-based explanation for the iteration and convergence of interactive teaching, such as co-teaching.

The EM process

Proposition 4.1. *Given the training dataset $\mathcal{D} = \{X_i = (x_i, y_i)\}_{i=1}^{N}$, which contains noisy samples, and assuming the samples are independent, we define the hidden variable $Z_c = 1$ to indicate that the c-th sample is a cleaner sample, meaning it has a lower loss value compared to noisy data. $Z = Z_c^f \cup Z_c^g = \{Z_c\}_{c=1}^{K}$ represents the set of hidden variables for all samples, and the corresponding latent distribution is denoted as $q(Z)$. The joint probability of $p(X_i, Z_c|\theta_f, \theta_g)$ is obtained by simultaneously updating neural networks f and g for X_i and Z_c. The logarithm likelihood of the observed data $\mathcal{D}$ has the following lower bound L:*

$$L(\theta_f, \theta_g, q) \equiv \sum_{i=1}^{N}\sum_{c=1}^{K}\left[q(Z_c)\log\frac{p(X_i, Z_c|\theta_f, \theta_g)}{q(Z_c)}\right], \tag{12}$$

the EM algorithm approximates the maximization of $\log p(\mathcal{D}|\theta_f, \theta_g)$ by maximizing this lower bound $L(\theta_f, \theta_g, q)$. Specifically, the EM iteration process for the interactive teaching paradigm is as follows:

E-step:

$$Q(\theta_f, \theta_g^{(t)}) = \mathbb{E}_{Z_c^f|\mathcal{D},\theta_f^{(t)},\theta_g}[\log p(Z_c^f, \mathcal{D}|\theta_f, \theta_g^{(t)})], \tag{13}$$

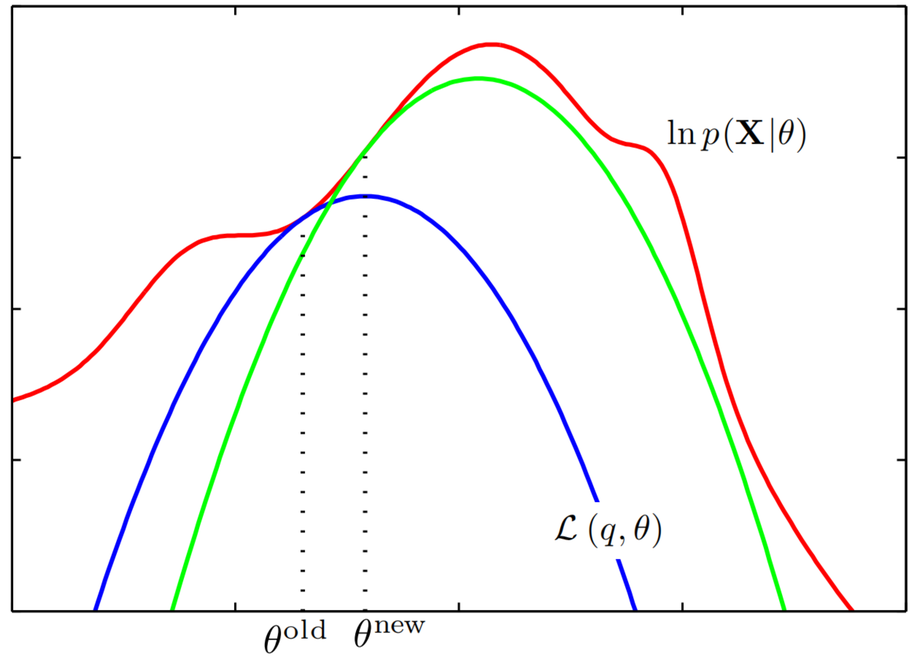
$$Q(\theta_f^{(t)}, \theta_g) = \mathbb{E}_{Z_c^g|\mathcal{D},\theta_f,\theta_g^{(t)}}[\log p(Z_c^g, \mathcal{D}|\theta_f^{(t)}, \theta_g)], \tag{14}$$

The subscript $Z_c^f|\mathcal{D}, \theta_f^{(t)}, \theta_g$ (or $Z_c^g|\mathcal{D}, \theta_f, \theta_g^{(t)}$) of the expectation represents the corresponding set of low-loss samples selected by network f (or g) and is used to update network g (or f).

M-step:

$$\theta_f^{(t+1)} = \arg\max_{\theta_f} Q(\theta_f^{(t)}, \theta_g), \tag{15}$$

$$\theta_g^{(t+1)} = \arg\max_{\theta_g} Q(\theta_f, \theta_g^{(t)}). \tag{16}$$



Solutions

◆ Since the convergence of the EM algorithm guarantees only local optima, while SAM can flatten the loss landscape and effectively alleviate local optima, favoring global optima, we incorporate SAM into the interactive teaching process, which referred as Sharpness Reduction Interactive Teaching (SRIT).

The Key steps

(1) For the first level, we start by screening the required low-loss dataset $\min_{\hat{\mathcal{D}}} \mathcal{L}(f_\theta, \mathcal{D})$.

(2) For the second level, we transfer the loss information to the counterpart model for SAM optimization: $\theta^* = \arg\min_{\hat{\theta}} \max_{\hat{\epsilon}} \mathcal{L}(g_{\theta+\epsilon}, \hat{\mathcal{D}})$.

The first step estimates the direction of the change in the network weights $\hat{\boldsymbol{\epsilon}}(\theta_f)$ and $\hat{\boldsymbol{\epsilon}}(\theta_g)$, and then computes $\boldsymbol{G}_f$ and $\boldsymbol{G}_g$,

$$\boldsymbol{G}_f = \nabla_{\theta_f} \mathcal{L}(f_\theta, \hat{\mathcal{D}}_g)|_{\theta_f + \hat{\boldsymbol{\epsilon}}(\theta_f)}, \boldsymbol{G}_g = \nabla_{\theta_g} \mathcal{L}(g_\theta, \hat{\mathcal{D}}_f)|_{\theta_g + \hat{\boldsymbol{\epsilon}}(\theta_g)}, \tag{17}$$

where $\hat{\boldsymbol{\epsilon}}(\theta_f) = \rho \frac{\nabla_\theta \mathcal{L}_{\hat{\mathcal{D}}_g}(f_\theta)}{\left\| \nabla_\theta \mathcal{L}_{\hat{\mathcal{D}}_g}(f_\theta) \right\|_2^2}, \hat{\boldsymbol{\epsilon}}(\theta_g) = \rho \frac{\nabla_\theta \mathcal{L}_{\hat{\mathcal{D}}_f}(g_\theta)}{\left\| \nabla_\theta \mathcal{L}_{\hat{\mathcal{D}}_f}(g_\theta) \right\|_2^2}$.

The second step involves updating the parameters of networks f and g based on the estimated gradients.

Algorithm & Experiments

Algorithm 1: Sharpness Reduction Interactive Teaching (SRIT)

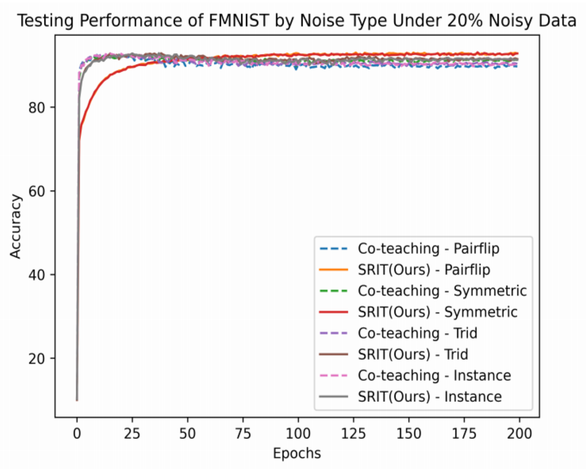
Input: Initial network parameters $\theta_{f_0}, \theta_{g_0}$, learning rate η, fixed parameter τ, iteration counts T_k and $T_{\max}$, maximum iteration count $N_{\max}$, pre-defined constant ρ.

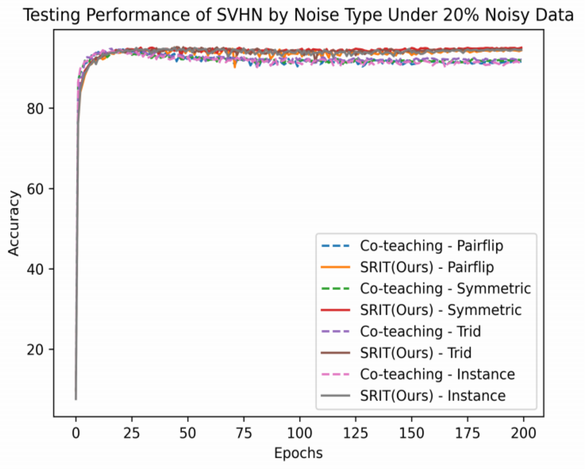
Output: Updated network parameters θ_f and θ_g.

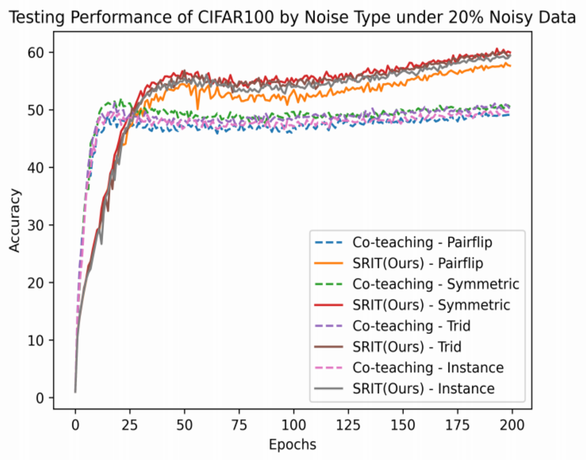
for $T = 1$ **to** $T_{\max}$ **do**
  Shuffle the training set $\mathcal{D}$ (noisy dataset);
  for $N = 1$ **to** $N_{\max}$ **do**
    Sample a mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
    Dual-level optimization: The first level of loss information exchange.
    Compute the loss of network f on $\bar{\mathcal{D}}$ and obtain $\hat{\mathcal{D}}_f$:
    $\hat{\mathcal{D}}_f = \arg\min_{\mathcal{D}':|\mathcal{D}'|\geq R(T)|\bar{\mathcal{D}}|} \mathcal{L}(f, \bar{\mathcal{D}})$; //sample $R(T)\cdot|\bar{\mathcal{D}}|$ small-loss instances;
    Compute the loss of network g on $\bar{\mathcal{D}}$ and obtain $\hat{\mathcal{D}}_g$:
    $\hat{\mathcal{D}}_g = \arg\min_{\mathcal{D}':|\mathcal{D}'|\geq R(T)|\bar{\mathcal{D}}|} \mathcal{L}(g, \bar{\mathcal{D}})$ //sample $R(T)\cdot|\bar{\mathcal{D}}|$ small-loss instances;
    Dual-level optimization : The second level of sharpness element exchange.
    Update $\hat{\boldsymbol{\epsilon}}(\theta_f) = \rho\frac{\nabla_\theta \mathcal{L}_{\hat{\mathcal{D}}_g}(f_\theta)}{\left\|\nabla_\theta \mathcal{L}_{\hat{\mathcal{D}}_g}(f_\theta)\right\|_2^2}$ and $\hat{\boldsymbol{\epsilon}}(\theta_g) = \rho\frac{\nabla_\theta \mathcal{L}_{\hat{\mathcal{D}}_f}(g_\theta)}{\left\|\nabla_\theta \mathcal{L}_{\hat{\mathcal{D}}_f}(g_\theta)\right\|_2^2}$;
    Compute the approximate gradient for network f: $\boldsymbol{G}_f = \nabla_{\theta_f}\mathcal{L}(f_\theta, \hat{\mathcal{D}}_g)|_{\theta_f+\hat{\boldsymbol{\epsilon}}(\theta_f)}$;
    Compute the approximate gradient for network g: $\boldsymbol{G}_g = \nabla_{\theta_g}\mathcal{L}(g_\theta, \hat{\mathcal{D}}_f)|_{\theta_g+\hat{\boldsymbol{\epsilon}}(\theta_g)}$;
    Update the network parameters of f using gradient descent: $\theta_f = \theta_f - \eta\boldsymbol{G}_f$;
    Update the network parameters of g using gradient descent: $\theta_g = \theta_g - \eta\boldsymbol{G}_g$;
  end
  Compute $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$.
end

Table 1: Test accuracy (%) on five datasets. The best results are highlighted in bold.

Noise type	Symmetric.		Pairflip.		Tridiagonal.		Instance.	
Noise ratio	20%	40%	20%	40%	20%	40%	20%	40%
				MNIST				
Co-teaching	97.50 ±0.06	94.96. ±0.07	95.49 ±0.11	91.54 ±0.15	96.61 ±0.06	92.76 ±0.09	95.90 ±0.05	91.23 ±0.18
SRIT	**99.42 ±0.03**	**99.19 ±0.03**	**99.35 ±0.02**	**98.14 ±0.07**	**99.47 ±0.03**	**98.75 ±0.05**	**99.43 ±0.02**	**98.03 ±0.10**
				CIFAR10				
Co-teaching	82.15 ±0.09	77.38 ±0.15	82.32 ±0.08	75.37 ±0.14	82.77 ±0.07	76.41 ±0.17	81.86 ±0.12	73.61. ±0.25
SRIT	**85.64 ±0.15**	**79.83 ±0.12**	**85.10 ±0.20**	**76.95 ±0.17**	**85.39 ±0.18**	**78.90 ±0.12**	**84.77 ±0.19**	**74.07 ±0.25**
				CIFAR100				
Co-teaching	50.21 ±0.23	42.40 ±0.16	48.27 ±0.11	34.74 ±0.13	50.32 ±0.19	38.78 ±0.16	49.74 ±0.18	38.57 ±0.12
SRIT	**59.66 ±0.16**	**50.57 ±0.21**	**57.16 ±0.10**	**35.82 ±0.16**	**59.07 ±0.15**	**42.27 ±0.22**	**59.66 ±0.16**	**40.36 ±0.18**
				FMNIST				
Co-teaching	91.13 ±0.09	87.99 ±0.09	89.83 ±0.10	85.44 ±0.12	90.42 ±0.07	86.09 ±0.09	90.27 ±0.12	85.63 ±0.13
SRIT	**92.68 ±0.10**	**88.77 ±0.13**	**92.76 ±0.09**	**89.25 ±0.11**	**91.44 ±0.11**	**89.97 ±0.09**	**91.44 ±0.09**	**86.02 ±0.21**
				SVHN				
Co-teaching	91.83 ±0.08	88.72 ±0.10	91.49 ±0.10	85.09 ±0.15	92.16 ±0.10	87.51 ±0.13	91.26 ±0.18	86.33 ±0.23
SRIT	**94.95 ±0.05**	**93.06 ±0.05**	**94.34 ±0.08**	**89.37 ±0.20**	**94.66 ±0.05**	**91.56 ±0.12**	**94.45 ±0.08**	**90.50 ±0.10**







Thank you!