"Improving De-raining Generalization via Neural Reorganization" Supplementary Material

Jie Xiao[†], Man Zhou[†], Xueyang Fu[*], Aiping Liu, Zheng-Jun Zha
University of Science and Technology of China, China
{ustchbxj,manman}@mail.ustc.edu.cn, {xyfu,aipingl,zhazj}@ustc.edu.cn

1. More details about synaptic consolidation

In our paper, we have mentioned that the Synaptic Consolidation module should include a submodule to concentrate on figuring out importance of synapses. In this section, we focus on the details of the way we adopt to evaluate the importance of synapses.

Most of deep learning based de-raining methods are capable of obtaining remarkable results by heuristically constructing a complicated neural network architecture in an end-to-end fashion. These methods regard the CNN as an encapsulated end-to-end mapping module to map the input image to its clean counterpart. Specifically, for the rain imaging process, it can be formulated as:

$$O = R + B, \tag{1}$$

where the O and B denote the rainy and the clean image respectively. The CNN based methods treat the rain removal as the mapping from input to output:

$$G = f(O; \theta), \tag{2}$$

where f indicates the CNN equipped de-raining architecture and θ refers to the parameter set of the network. Besides, the loss function is employed to evaluate the difference between the output and the ground truth. The training of de-raining network corresponds to minimize the objective loss function given data from training set, recorded as:

$$\begin{aligned} &\min \quad \mathcal{L}(G, B) \\ =&\min_{\theta} \quad \mathcal{L}(f(O;\theta), B), \end{aligned} \tag{3}$$

where $\mathcal{L}$ indicates the conventional loss (*e.g.* MSE [1]) to train de-raining network. For simplicity, Eq. (3) is rewritten as:

$$\min_{\theta} \quad \mathcal{L}(O, B; \theta). \tag{4}$$

We consider the situation that training the de-raining network on a sequence of datasets, whose total length is recorded as N, with our Neural Reorganization. The parameter set of de-raining architecture f is denoted as θ^n when the training of network on dataset n is finished. The continual two rainy image sets are denoted as $X^n, X^{n+1} (0 \leq n < N-1)$ and their clean counterparts are remarked as $Y^n, Y^{n+1} (0 \leq n < N-1)$ respectively. For $x^n \in X^n, y^n \in Y^n$, suppose x^n is random variable, which is independently and identically distributed to $\mathbb{P}^n$, and y^n is the corresponding clean image. If x_n is fed into network, the degradation of performance on dataset n introduced by the training of network on dataset $n+1$ is evaluated by:

$$\begin{aligned} &\text{Distance}(f(x^n;\theta), f(x^n;\theta^n)) \\ \triangleq\ &\left|\mathcal{L}(f(x^n;\theta), y^n) - \mathcal{L}(f(x^n;\theta^n), y^n)\right| \\ =\ &\left|\mathcal{L}(x^n, y^n;\theta) - \mathcal{L}(x^n, y^n;\theta^n)\right|, \end{aligned} \tag{5}$$

where $|\cdot|$ denotes absolute value operator, $\triangleq$ means definition, the function Distance measures the distance between $f(x^n;\theta)$ and $f(x^n;\theta^n)$, $\mathcal{L}$ represents conventional loss used by training de-raining network. It is noteworthy that we use θ to refer to the trainable parameters, which shall evolve to θ^{n+1} when the training procedure is finished on dataset $n+1$. The change of parameter θ^n when model is trained on the new dataset $n+1$ is denoted by $\Delta\theta^n$ whose mathematical form is

$$\Delta\theta^n = \theta - \theta^n. \tag{6}$$

To evaluate $\text{Distance}(f(x^n;\theta), f(x^n;\theta^n))$, we take the Taylor expansion of $\mathcal{L}(x^n, y^n;\theta)$ at point θ^n, which is an infinite sum of terms that are expressed in the form of target function's derivatives at a single point:

$$\begin{aligned} &\mathcal{L}(x^n, y^n;\theta^n + \Delta\theta^n) = \mathcal{L}(x^n, y^n;\theta^n) \\ &+ \left(\nabla_\theta \mathcal{L}(x^n, y^n;\theta)\right)^T|_{\theta=\theta^n} \cdot \Delta\theta^n + O(\|\Delta\theta^n\|^2). \end{aligned} \tag{7}$$

Combining Eq. (5) and Eq. (7), we can get the approximation:

$$\begin{aligned} &\text{Distance}(f(x;\theta), f(x^n;\theta^n)) \\ \approx\ &\left|\left(\nabla_\theta \mathcal{L}(x^n, y^n;\theta)\right)^T|_{\theta=\theta^n} \cdot \Delta\theta^n\right|. \end{aligned} \tag{8}$$

[*]Corresponding author. [†]Co-first authors contributed equally

Clearly, $|\nabla_\theta \mathcal{L}(x^n, y^n; \theta)|$ measures the importance of parameters. Specifically, the larger element of $|\nabla_\theta \mathcal{L}(x^n, y^n; \theta)|$ means the corresponding parameter (synapse) is more influential to previous tasks. Hence, we define Ω to evaluate the importance of synapse over dataset n, whose mathematical form is:

$$\Omega = \mathop{\mathbb{E}}_{x \sim \mathbb{P}^n} \left[\left| \frac{\partial \mathcal{L}(f(x;\theta), y)}{\partial \theta} \right|_{\theta = \theta^n} \right]. \tag{9}$$

References

[1] J. Xu, W. Zhao, P. Liu, and X. Tang. Removing rain and snow in a single image using guided filter. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, volume 2, pages 304–307, 2012. 1