MaxUp: A Simple Way to Improve Generalization of Neural Network Training

Seungmin Lee (profile2697@gmail.com; 2020-20866),
Dept. of Electrical and Computer Engineering, Seoul National University

1. Introduction

This paper proposes a simple training strategy called MaxUp. The authors claim that MaxUp is embarrassingly simple and easy to apply while it still enhances performance. The central intuition behind MaxUp is similar to that of adversarial training, which is regularizing networks by inducing them to lie on a smooth loss landscape. MaxUp implements the smoothness regularization by generating a set of randomly augmented samples to calculate the worst-case loss for a given image and letting networks minimize the worst-case loss.

2. Method

Empirical Risk Minimization (ERM) is the standard scheme for training a regular neural network:

$$\min_{\theta} \mathbb{E}_{x \sim \mathcal{D}_n} \left[L(x; \theta) \right], \tag{1}$$

where θ and L denote the parameters of the network and the loss function, respectively, and $\mathcal{D}_n$ indicates the dataset while x is a sampled data from $\mathcal{D}_n$. While ERM is simple, the authors argue that ERM often incurs networks' overfitting.

The proposed method, MaxUp, aims to alleviate such overfitting by first generating a set of randomly augmented data, then minimizing the maximum loss over this set:

$$\min_{\theta} \mathbb{E}_{x \sim \mathcal{D}_n} \left[\max_{i \in [m]} L(x'_i, \theta) \right]. \tag{2}$$

Compared to Equation (1), Equation (2) now has the added $\max$ term, as well as the x'_i term instead of x, where x'_i denotes the i-th random augmentation. Thus, the objective now is to find the augmentation of x with the maximum loss, then minimize this loss.

3. Experiments and Thoughts

Table 1 in the main paper shows the Top-1 and Top-5 accuracies on the validation of ImageNet with ResNet. The table compares MaxUp + CutMix with other previous methods, such as Dropout, Mixup [?], and CutMix [?](without MaxUp). A similar trend is shown in Table 2, which shows the performance of MaxUp + CutMix compared to solely CutMix on ImageNet, using more recent architectures such as EfficientNet [?]. The authors also show performance improvements in CIFAR10 and CIFAR100 datasets.

From my personal point of view, the results do not seem too impressive because they always improve upon the results of CutMix (without MaxUp) by a very small margin. For example, CutMix seems to improve the Top-1 accuracy of the Vanilla setting by around 2.3% points, while MaxOut + CutMix only improves upon the Vanilla by 2.6% points. If we assume that MaxUp and CutMix are orthogonal methods, it seems like MaxUp only has a performance improvement of around 0.3% points. I am not sure why the authors decided to use CutMix as the base method in all experiments, and it would be interesting to see results without CutMix. Furthermore, I am concerned about the training time of this method. If we have 10 different augmentations for each sample, it means that each backpropagated sample will require 10 forward passes in order to find the maximum loss. I do not feel that this is practical, or even worth it, given that the performance improvement is quite marginal.

References