Generalized Cross Entropy Loss for Noisy Labels

Zhilu Zhang and Mert R. Sabuncu
Cornell University

Motivation

Deep neural networks:

- Often need lots of clean labeled data - can be expensive to obtain
- Can overfit to noisy labels [Zhang et al. 2016]

Symmetric Loss

- A loss function is *symmetric* if

$$\sum_{j=1}^{c} \mathcal{L}(f(\boldsymbol{x}), j) = const, \quad \forall x \in \mathcal{X}, \forall f$$

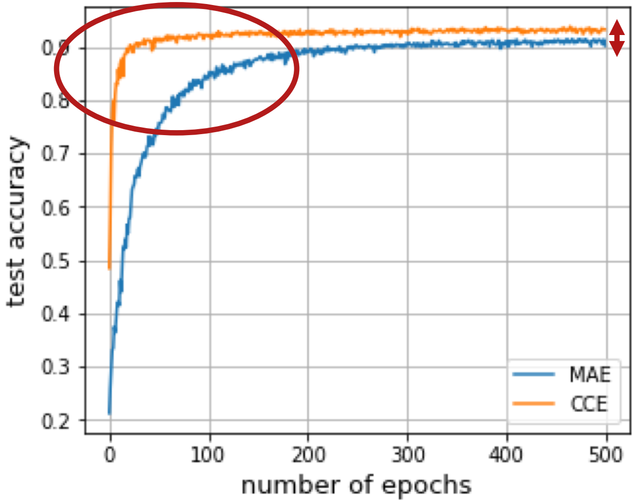
- *Symmetric* loss can be tolerant to noisy labels [Ghosh et al. 2017]
- MAE for classification with probabilistic outputs is symmetric

$$\mathcal{L}_{MAE}(f(\boldsymbol{x}), \mathbf{e}_j) = ||\mathbf{e}_j - f(\boldsymbol{x})||_1 = 2 - 2f_j(\boldsymbol{x}).$$

Limitations of MAE

- MAE is noise-robust but can converge to lower accuracy

Much slower convergence



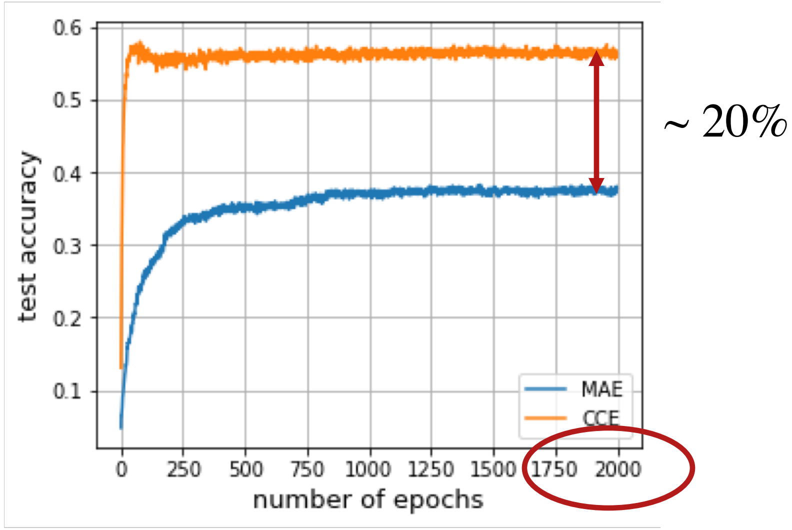
Slight gap in test accuracy

ResNet on CIFAR-10

Limitations of MAE

- MAE is noise-robust but can converge to lower accuracy

Using MAE, the highest accuracy achieved is 38.29% in 2000 epochs, and CCE achieved better performance after 7 epochs!



ResNet on CIFAR-100

Generalized Cross Entropy (Lq Loss)

CCE

- Good convergence, but prone to label noise

MAE

- More noise robust, but bad convergence

Use the Box-Cox Transformation to combine them

$$\mathcal{L}_q(f(\boldsymbol{x}), \boldsymbol{e}_j) = \frac{(1 - f_j(\boldsymbol{x})^q)}{q}$$

Generalized Cross Entropy (Lq Loss)

$$\mathcal{L}_q(f(\boldsymbol{x}), \boldsymbol{e}_j) = \frac{(1 - f_j(\boldsymbol{x})^q)}{q}$$

CCE | MAE

$q = 0$ | $q \in [0,1]$ | $q = 1$

- Lq loss has bounded sum of losses for non zero q

$$\frac{c - c^{(1-q)}}{q} \leq \sum_{j=1}^{c} \frac{(1 - f_j(\boldsymbol{x})^q)}{q} \leq \frac{c-1}{q}$$

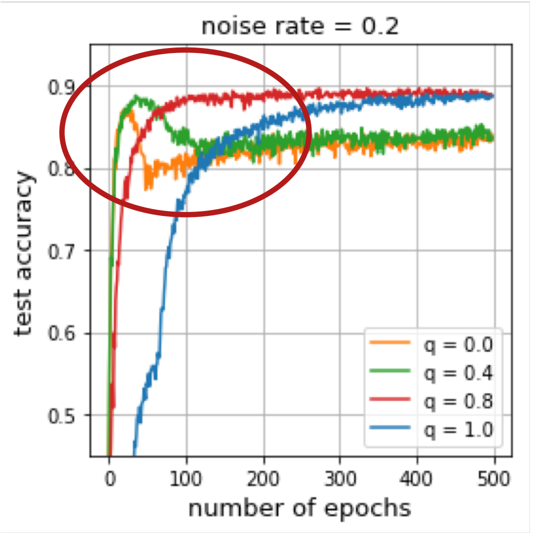
- The tighter the bound, the more noise robust the Lq loss

Generalized Cross Entropy (Lq Loss)

$$\mathcal{L}_q(f(\boldsymbol{x}), \boldsymbol{e}_j) = \frac{(1 - f_j(\boldsymbol{x})^q)}{q}$$

CCE ⟷ MAE

$q = 0$ $q \in [0,1]$ $q = 1$



ResNet on CIFAR-10

Truncated Lq Loss

- Propose the truncated Lq loss

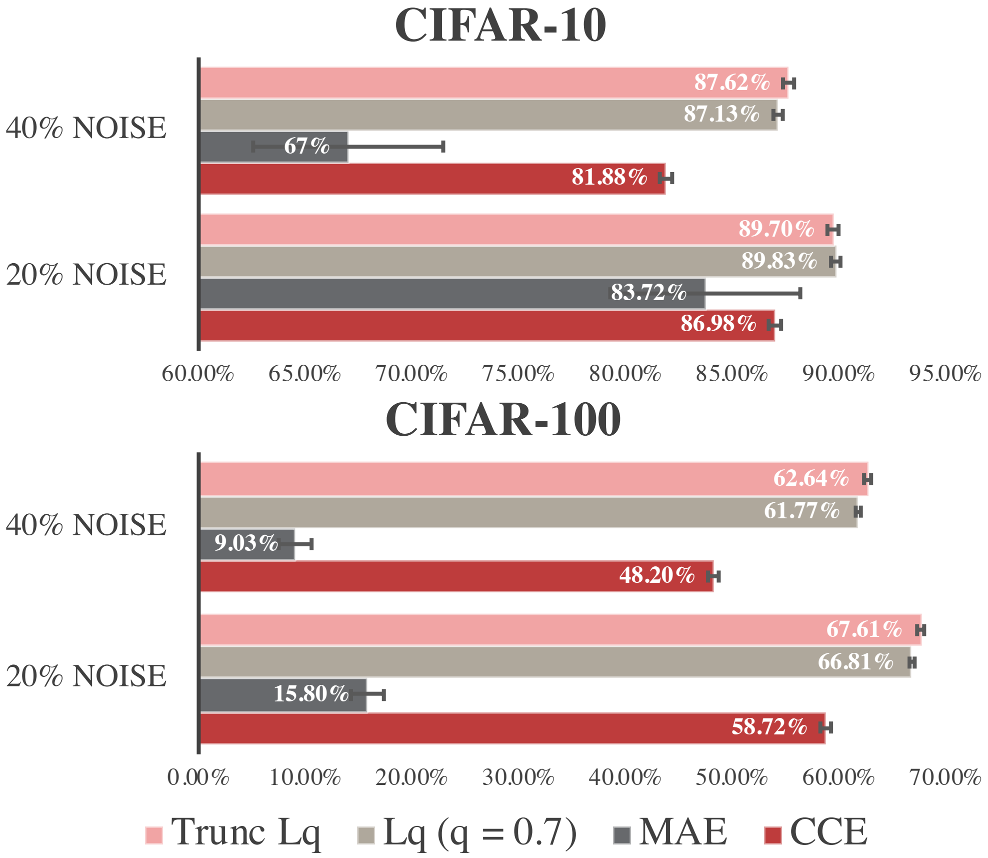
$$\mathcal{L}_{trunc}(f(\boldsymbol{x}), \boldsymbol{e}_j) = \begin{cases} \mathcal{L}_q(k) & \text{if } f_j(\boldsymbol{x}) \leq k \\ \mathcal{L}_q(f(\boldsymbol{x}), \boldsymbol{e}_j) & \text{if } f_j(\boldsymbol{x}) > k \end{cases}$$

- Often has tighter bound
- Use alternative convex search algorithm for optimization

Experiments

- ResNet on CIFAR-10, CIFAR-100 and FASHION-MNIST with synthetic noise

- Consistent improvements over CCE and MAE



- Thank you very much for your attention!
- Hope to see you at Poster #101