Robust Losses

Sixu Li

1 Preliminary

Given a C-class dataset $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^n$, with $\boldsymbol{x_i} \in \mathcal{X} \subset \mathbb{R}^d$ denoting a sample and $y_i \in \mathcal{Y} = \{1, \cdots, C\}$ its associated label. For each sample $\boldsymbol{x_i}$, a classifier $f(\boldsymbol{x_i})$(DNNs here) computes its probability of each label $j \in 1, \cdots, C$: $p(j|\boldsymbol{x_i}) = \frac{\exp(\boldsymbol{z}_{ij})}{\sum_{k=1}^C \exp(\boldsymbol{z}_{ik})}$,where $\boldsymbol{z}_i$ are the logits on sample $\boldsymbol{x_i}$. We denote the ground-truth distribution over labels for sample $\boldsymbol{x_i}$ by $q(j|\boldsymbol{x_i})$, and $\sum_{j=1}^C q(j|\boldsymbol{x_i}) = 1$. Consider the case of a single ground-truth label y_i, then $q(y_i|\boldsymbol{x_i}) = 1$ and $q(j|\boldsymbol{x_i}) = 0$ for all $j \neq y_i$. Then categorical cross entropy(CCE) loss and mean absolute error(MAE) loss for sample $\boldsymbol{x_i}$ can be writen as:

$$\mathcal{L}_{\text{CCE}}(\boldsymbol{x_i}) = -\sum_{j=1}^{C} q(j|\boldsymbol{x_i}) \log p(j|\boldsymbol{x_i}) = -\log p(y_i|\boldsymbol{x_i}) \tag{1}$$

$$\mathcal{L}_{\text{MAE}}(\boldsymbol{x_i}) = \sum_{j=1}^{C} |p(j|\boldsymbol{x_i}) - q(j|\boldsymbol{x_i})| = 2(1 - p(y_i|\boldsymbol{x_i})) \tag{2}$$

2 Summary of Papers

- **"Robust Loss Functions under Label Noise for Deep Neural Networks"[1]**
  Key Results:
  (i). Define "Symmetric Losses":
  We call a loss function L symmetric if it satisfies, for some constant C,

$$\sum_{i=1}^{k} \mathcal{L}(f(x), i) = C, \quad \forall x \in \mathcal{X}, \forall f \tag{3}$$

  For these loss functions, we have,

$$\sum_{i=1}^{k} \mathcal{L}(f(x), e_i) = \begin{cases} \sum_{i=1}^{k} \log \frac{1}{f_i(x)}, & \text{CCE} \\ \sum_{i=1}^{k} (2 - 2f_i(x)) = 2k - 2, & \text{MAE} \\ k\|f_i(x)\|_2^2 + k - 2, & \text{MSE} \end{cases} \tag{4}$$

Thus, among these, only MAE satisfies symmetry condition given by Eq.(3).

(ii). Theoretically proved that Symmetric Losses are robust to label noise with a little additional assumptions and verified it by empirical experiments

(iii). Use empirical results showing that Bounded Losses are more robust to unbounded losses.

- **"Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels"[5]**

  Key Results:

  (i). Give an explanation of robustness of MAE and noise-sensitiveness of CCE.

  Notations: Let $\mathcal{X} \subset \mathbb{R}^d$ be the feature space and $\mathcal{Y} = 1, \cdots, c$ be the label space, $(\boldsymbol{x_i}, y_i) \in (\mathcal{X}, \mathcal{Y})$. Denote classifier $f(\boldsymbol{x}; \boldsymbol{\theta}) : \mathcal{X} \to \mathcal{R}^c$ and f_j denotes the j'th element of f.

  Let's look at the gradient of the loss functions:

$$\sum_{i=1}^{n} \frac{\partial \mathcal{L}(f(\boldsymbol{x_i}; \boldsymbol{\theta}), y_i)}{\partial \boldsymbol{\theta}} = \begin{cases} \sum_{i=1}^{n} -\frac{1}{f_{y_i}(\boldsymbol{x_i}; \boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} f_{y_i}(\boldsymbol{x_i}; \boldsymbol{\theta}), & \text{for CCE} \\ \sum_{i=1}^{n} -\nabla_{\boldsymbol{\theta}} f_{y_i}(\boldsymbol{x_i}; \boldsymbol{\theta}), & \text{for MAE} \end{cases} \tag{5}$$

  In this paper, they claims that, in CCE, $\frac{1}{f_{y_i}(\boldsymbol{x_i}; \boldsymbol{\theta})}$ can be viewed as the weight put on each example when updating the gradient. Therefore, this means that, when training with CCE, more emphasis is put on difficult examples. This implicit weighting scheme is desirable for training with clean data, but can cause overfitting to noisy labels. Conversely, MAE treats every sample equally, which makes it more robust to noisy labels but meanwhile slower convergence rate. Moreover, without the implicit weighting scheme to focus on challenging samples, the stochasticity involved in the training process can make learning difficult. As a result, classification accuracy might suffer.

  My Question: Is the perspective that viewing $\frac{1}{f_{y_i}(\boldsymbol{x_i}; \boldsymbol{\theta})}$ as the weight put on each example when updating the gradient correct?

  In my opinion, large $\frac{1}{f_{y_i}(\boldsymbol{x_i}; \boldsymbol{\theta})}$ does not guarantee that sample $\boldsymbol{x_i}$ has large contribution to the gradient updating because $\nabla_{\boldsymbol{\theta}} f_{y_i}(\boldsymbol{x_i}; \boldsymbol{\theta})$ also contains information that $(\boldsymbol{x_i}, y_i)$ provides. It is possible that for large $\frac{1}{f_{y_i}(\boldsymbol{x_i}; \boldsymbol{\theta})}$, it corresponds to small $\nabla_{\boldsymbol{\theta}} f_{y_i}(\boldsymbol{x_i}; \boldsymbol{\theta})$. Similarly, the explanation for MAE also should be doubted.

  (ii). They proposed a generalized version of MAE and CCE:

$$\mathcal{L}_q(f(\boldsymbol{x}, \boldsymbol{e_j})) = \frac{(1 - f_j(\boldsymbol{x})^q)}{q} \tag{6}$$

  Using L'Hopital's rule, it can be shown that the proposed loss function is equivalent to CCE for $\lim_{q \to 0} \mathcal{L}_q(f(\boldsymbol{x}, \boldsymbol{e_j}))$, and becomes MAE when $q = 1$. This means that $\mathcal{L}_q$ utilizes

hyperparameter q to control the trade-off between fast convergence of CCE and noise-robustness of MAE.

Besides, in order to let $\mathcal{L}_q$ to have a tight bound, they proposed Truncated $\mathcal{L}_q$ loss:

$$\mathcal{L}_{\text{trunc}}(f(\boldsymbol{x}, \boldsymbol{e_j})) = \begin{cases} \mathcal{L}_q(k), & \text{if } f_j(\boldsymbol{x}) \leq k \\ \mathcal{L}_q(f(\boldsymbol{x}, \boldsymbol{e_j})), & \text{if } f_j(\boldsymbol{x}) > k \end{cases} \tag{7}$$

In this way, if the softmax output for the provided label is below a threshold, truncated $\mathcal{L}_q$ loss becomes a constant. Thus, the loss gradient is zero for that sample, and it does not contribute to learning dynamics.

Drawback: this process is equivalent to drop out those "hard" samples, which, in some senses, filter out those noisy samples. However, it will also ignore those challenging but clean data. Therefore, how to distinguish "hard" but clean samples and noisy samples is an important problem to think about.

- **"IMAE for Noise-Robust Learning: Mean Absolute Error Does Not Treat Examples Equally and Gradient Magnitude's Variance Matters"[2]**

  Prliminaries:

  Training set $\boldsymbol{X} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^N$, training sample $\boldsymbol{x_i} \in \mathbb{R}^D$ and its annotated class $y_i \in \{1, 2, \cdots C\}$. Let f_θ be a deep neural network, which transforms $\boldsymbol{x_i}$ to a representation $\boldsymbol{f_i} = f_\theta(\boldsymbol{x_i}) \in \mathbb{R}^E$.The linear classifier follows the output embeddings f_θ and is composed of one fully connected(FC) layer, one softmax layer and one loss layer. The FC layer can be represented as $\boldsymbol{z_i} = \boldsymbol{W}^T \boldsymbol{f_i} \in \mathbb{R}^C$, where $\boldsymbol{W} = [\boldsymbol{w_1}, \boldsymbol{w_2}, \cdots, \boldsymbol{w_C}] \in \mathbb{R}^{E \times C}$. $\boldsymbol{z_{ij}} = \boldsymbol{w_j^T} \boldsymbol{f_i}$ is a logit that indicates the compatibility between sample $\boldsymbol{x_i}$ and class j. A softmax function is given by,

  $$p(j|\boldsymbol{x_i}) = \frac{\exp(\boldsymbol{z_{ij}})}{\sum_{m=1}^C \exp(\boldsymbol{z_{ij}})} \tag{8}$$

  where $p(j|\boldsymbol{x_i})$ is the probability of sample $\boldsymbol{x_i}$ being predicted by class j.

  Key Results:

  (i). They define gradient magnitude w.r.t logit vector $\boldsymbol{z}$ and the gradients' variance(See derivation in the original paper)

  $$\omega_{\text{CCE}}(\boldsymbol{x_i}) = \|\frac{\partial L_{\text{CCE}}(\boldsymbol{x_i})}{\partial \boldsymbol{z_i}}\|_1 = 2(1 - p(y_i|\boldsymbol{x_i})) \tag{9}$$

  $$\omega_{\text{MAE}}(\boldsymbol{x_i}) = \|\frac{\partial L_{\text{MAE}}(\boldsymbol{x_i})}{\partial \boldsymbol{z_i}}\|_1 = 4p(y_i|\boldsymbol{x_i})(1 - p(y_i|\boldsymbol{x_i})) \tag{10}$$

  Assuming that samples' probabilities $p(y_i|\boldsymbol{x_i})$ are uniformly distributed, then the gradi-

ents' variance of MAE over training data points can be writen as,

$$\sigma_{\text{MAE}} = \int_0^1 \omega_{\text{MAE}}^2(p)dp - (\int_0^1 \omega_{\text{MAE}}(p)dp)^2 \tag{11}$$

They treat gradient magnitude as weight on each sample, i.e, if one sample's gradient is larger, its impact is larger during gradient back-propagation.
Based on this consideration, they provide two explanations of MAE's properties:
1. MAE emphasizes on uncertain examples($p(y_i|\boldsymbol{x_i})$ is closer to 0.5) and those samples with high-probability and low-probability have smaller weight. While low-probability ones are highly likely to be noisy as a model improves during training, therefore, MAE is noise-robustness.
2. MAE's gradient magnitude's variance over data points is only 0.09(too small). As a consequence, the impact ratio of one example versus another is too small. Therefore, the majority contribute almost equally. Therefore, MAE generally underfits to training data.
My Question: Is the gradient magnitudes truly reflect the samples' impact on the gradient updating? Is the L_1 norm of gradients meaningful?
My suspicion on this problem: For $\forall$ loss function $\mathcal{L}$, following the previous notation, when training on data point $(\boldsymbol{x_i}, y_i)$, now we consider the gradients on parameters W. By chain rule, we have,

$$\begin{aligned}\frac{\partial \mathcal{L}(\boldsymbol{x_i})}{\partial \boldsymbol{W}} &= \frac{\partial \mathcal{L}(\boldsymbol{x_i})}{\partial \boldsymbol{z_i}} \frac{\partial \boldsymbol{z_i}}{\partial \boldsymbol{W}} \\ &= \frac{\partial \mathcal{L}(\boldsymbol{x_i})}{\partial \boldsymbol{z_i}} \boldsymbol{f_i} \in \mathbb{R}^{E\times C}\end{aligned} \tag{12}$$

Following original paper's definition, if we define L_1 norm of gradients to be gradients' magnitude, which measures the impact of each training sample on gradient updating. Then, let's consider two data point $(\boldsymbol{x_1}, y_1)$ and $(\boldsymbol{x_2}, y_2)$. If we have $\|\frac{\partial \mathcal{L}(\boldsymbol{x_1})}{\partial \boldsymbol{z_1}}\|_1 > \|\frac{\partial \mathcal{L}(\boldsymbol{x_2})}{\partial \boldsymbol{z_2}}\|_1$, under this paper's conclusion, the impact of sample $\boldsymbol{x_1}$ is larger than $\boldsymbol{x_2}$ during gradient back-propagation. However, it doesn't guarantee that $\|\frac{\partial \mathcal{L}(\boldsymbol{x_1})}{\partial \boldsymbol{W}}\|_1 > \|\frac{\partial \mathcal{L}(\boldsymbol{x_2})}{\partial \boldsymbol{W}}\|_1$.
So, one importent problem to consider, how can measure training samples' impact on gradient updating?
(ii). In order to solve the problem that MAE's gradient magnitude's variance over data points is too small. They define IMAE that transforms MAE's weighting scheme non-linearly:

$$\omega_{\text{IMAE}}(\boldsymbol{x_i}) = \exp(Tp(y_i|\boldsymbol{x_i}))(1 - p(y_i|\boldsymbol{x_i})) \tag{13}$$

And in back-propagation, they simply scale the gradient w.r.t logits as follows:

$$\frac{\partial \mathcal{L}_{\mathrm{IMAE}(\boldsymbol{x_i})}}{\partial \boldsymbol{z_i}} = \frac{\partial \mathcal{L}_{\mathrm{MAE}(\boldsymbol{x_i})}}{\partial \boldsymbol{z_i}} \frac{\omega_{\mathrm{IMAE}}(\boldsymbol{x_i})}{\omega_{\mathrm{MAE}}(\boldsymbol{x_i})} \tag{14}$$

(Figure that compares the gradients' variance is shown in the original paper.)
Drawback: By looking at the Figure 4 in the original paper, we can find out that the convergence rate of IMAE is the same as MAE, which is still a problem.
My Question: When trainging on clean data point $(\boldsymbol{x_i}, y_i)$, let's look at the following two gradient on the logit $\boldsymbol{z_{ij}}$:

$$\frac{\partial \mathcal{L}_{\mathrm{CCE}}(\boldsymbol{x_i})}{\partial \boldsymbol{z_{ij}}} = \begin{cases} p(y_i|\boldsymbol{x_i}) - 1, & j = y_i \\ p(j|\boldsymbol{x_i}), & j \neq y_i \end{cases} \tag{15}$$

$$\frac{\partial \mathcal{L}_{\mathrm{MAE}}(\boldsymbol{x_i})}{\partial \boldsymbol{z_{ij}}} = \begin{cases} 2p(y_i|\boldsymbol{x_i})(p(y_i|\boldsymbol{x_i}) - 1), & j = y_i \\ 2p(y_i|\boldsymbol{x_i})p(j|\boldsymbol{x_i}), & j \neq y_i \end{cases} \tag{16}$$

We can observe that when $j = y_i$, both $\frac{\partial \mathcal{L}_{\mathrm{CCE}}(\boldsymbol{x_i})}{\partial \boldsymbol{z_{ij}}}$ and $\frac{\partial \mathcal{L}_{\mathrm{MAE}}(\boldsymbol{x_i})}{\partial \boldsymbol{z_{ij}}}$ are less than 0 and when $j \neq y_i$, they are all greater than zero. Therefore, is the sign of gradients meaningful?
The expressions of these two gradients when training on noise label $(\boldsymbol{x_i}, \hat{y}_i)$ can be extended easily, but need to take into consideration.

- **"Symmetric Cross Entropy for Robust Learning with Noisy Labels"[3]**
  Key Results:
  (i). They did a class-based experiment, which tracks the test accuracy of each class along the training procedure(see the experimental results in the original paper). In this experiment, they found out that DNN learning with CE can be **class-biased:** some classes("easy" classes) are easy to learn and converge faster than other classes("hard" classes). And this phenomenon is amplified when training labels are noisy: while easy classes already overfit to noisy labels, hard classes still suffer from significant **under learning**(class accuracy significantly lower than clean label setting). They claim that it appears that the under learining of hard classes is a major cause for the overall performance degradation.(The experiment phenomenons to prove their claim are shown in the original paper).
  My Question: They only do this experiment on data set CIFAR-10 with 40% symmetric noise, hence, is this conclusion true on other data set with different kinds of noise?
  (ii). Inspired by the symmetric KL-divergence, they proposed a Symmetric Cross Entropy(SCE) as,

$$SCE = CE + RCE = H(q, p) + H(p, q) \tag{17}$$

And we can write loss function Reverse Cross Entropy(RCE) as,

$$\mathcal{L}_{\mathrm{RCE}}(\boldsymbol{x_i}) = -\sum_{j=1}^{C} p(j|\boldsymbol{x_i}) \log q(j|\boldsymbol{x_i}) \tag{18}$$

And define,

$$\mathcal{L}_{\mathrm{SL}} = \alpha\mathcal{L}_{\mathrm{CE}} + \beta\mathcal{L}_{\mathrm{RCE}} \tag{19}$$

Because $q(j|\boldsymbol{x_i})$ is an one-hot vector, in order to let RCE valid, they define $\log 0$ to be a negative number A. And similar to [1], they prove that RCE is noise-robustness under a little assumption.

And let's look at the gradients of SL on the logits $\boldsymbol{z_{ij}}$ when training on clean data point $(\boldsymbol{x_i}, y_i)$.

$$\frac{\partial\mathcal{L}_{\mathrm{SL}}(\boldsymbol{x_i})}{\partial \boldsymbol{z_{ij}}} = \begin{cases} \frac{\partial\mathcal{L}_{\mathrm{CE}}(\boldsymbol{x_i})}{\partial \boldsymbol{z_{ij}}} + Ap(y_i|\boldsymbol{x_i})(1 - p(y_i|\boldsymbol{x_i})), & j = y_i \\ \frac{\partial\mathcal{L}_{\mathrm{CE}}(\boldsymbol{x_i})}{\partial \boldsymbol{z_{ij}}} - Ap(y_i|\boldsymbol{x_i})p(j|\boldsymbol{x_i}), & j \neq y_i \end{cases} \tag{20}$$

We can take a look at the experimental results of SL in the original paper.

My Question:

(1). Look at their explanation of this gradient, I think it does not make sense.

(2). All of these paper only explain the behavior of these gradients when training on clean data, however, they do not consider the form of these gradients when training on noisy label.(Write out those gradients when training on noisy data and take a look). And this question relates to the meaning of gradients' sign on each component.

- **"$\mathcal{L}_{\mathrm{DMI}}$: A Novel Information-theoretic Loss Function for Training Deep Nets Robust to Label Noise"** [4]

  Preliminaries:

  Denote the set of classes by $\mathcal{C}$ and the size of $\mathcal{C}$ is C. Also, denote the domain of datapoints by $\mathcal{X}$. A classifier is denoted by $h : \mathcal{X} \to \Delta_{\mathcal{C}}$, where $\Delta_{\mathcal{C}}$ is the set of all possible distributions over $\mathcal{C}$. h represents a randomized classifer such that given $x \in \mathcal{X}$, $h(x)_c$ is the probability that h maps x into class c. Note that fixing the input x, the randomness of a classifier is independent of everything else.

  There are N datapoints $\{x_i\}_{i=1}^N$. For each datapoint x_i, there is an unknown ground truth $y_i \in \mathcal{C}$. We assume that there is an unkown prior distribution $Q_{X,Y}$ over $\mathcal{X} \times \mathcal{C}$ such that $\{(x_i, y_i)\}_{i=1}^N$ are i.i.d. samples drawn from $Q_{X,Y}$ and

$$Q_{X,Y}(x, y) = Pr[X = x, Y = y] \tag{21}$$

And we Denote matrix format of joint distribution over X and Y as,

$$\boldsymbol{Q}_{X,Y} = \big[Pr[X = x_i, Y = y_i]\big] \tag{22}$$

Besides, define noise transition matrix as,

$$\boldsymbol{T}_{Y\to\widetilde{Y}} = \big[Pr[\widetilde{Y} = \widetilde{y}|Y = y]\big]_{C\times C} \tag{23}$$

Key Results:

(1) Probelm of Shannon mutual information and new proposed determinant based mutual information.

For every two random variables W_1 and W_2, Shannon mutual information defined as

$$\mathbf{MI}(W_1, W_2) \triangleq \sum_{w_1,w_2} Pr[W_1 = w_1, W_2, w_2] \log \frac{Pr[W_1 = w_1, W_2 = w_2]}{Pr[W_1 = w_1]Pr[W_2 = w_2]} \tag{24}$$

Properties of $\mathbf{MI}(W_1, W_2)$:

(i). Non-negative.

(ii). Symmetric, i.e., $\mathbf{MI}(W_1, W_2) = \mathbf{MI}(W_2, W_1)$.

(iii). Information-monotonicity:

For all random variables W_1, W_2, W_3, when W_3 is less informative for W_2 than W_1, i.e., W_3 is independent of W_2 conditioning on W_1, then,

$$\mathbf{MI}(W_3, W_2) \leq \mathbf{MI}(W_1, W_2) \tag{25}$$

Based on Shannon mutual information, a performance measure for a classifer h can be naturally defined. High quality classifier's output $h(X)$ should have high mutual information with the ground truth category Y. Thus, a classifier h's performance can be measured by $\mathbf{MI}(h(X), Y)$.

Problem for $\mathbf{MI}(W_1, W_2)$ when training on noisy labels:

$$\forall h, h', \mathbf{MI}(h(X), Y) > \mathbf{MI}(h'(X), Y) \not\Leftrightarrow \mathbf{MI}(h(X), \widetilde{Y}) > \mathbf{MI}(h'(X), \widetilde{Y}) \tag{26}$$

In order to solve this problem, we proposed a new measure called Determinant based Mutual Information(**DMI**).

Definition:

Given two discrete random variables W_1, W_2, we define the Determinant based Mutual Information between W_1 and W_2 as

$$\mathbf{DMI}(W_1, W_2) = |\det(\mathbf{Q}_{W_1,W_2})| \tag{27}$$

where $\mathbf{Q}_{W_1,W_2}$ is the matrix format of the joint distribution over W_1 and W_2.
Properties of **DMI**:
(i). Non-negative.
(ii). Symmetric.
(iii). Information-monotonicity.
(iv). Relative Invariance (See the Proof in original paper):
For random variables W_1, W_2, W_3, when W_3 is less informative for W_2 than W_1, i.e., W_3 is independent of W_2 conditioning W_1,

$$\mathbf{DMI}(W_2, W_3) = \mathbf{DMI}(W_2, W_1)|\det(\mathbf{T}_{W_1 \to W_3})| \tag{28}$$

where $\mathbf{T}_{W_1 \to W_3}$ is the matrix format of

$$T_{W_1 \to W_3}(w_1, w_3) = Pr[W_3 = w_3 | W_1 = w_1] \tag{29}$$

Based on property (iv), we have,

$$\forall h, h', \mathbf{DMI}(h(X), Y) > \mathbf{DMI}(h'(X), Y) \Leftrightarrow \mathbf{DMI}(h(X), \widetilde{Y}) > \mathbf{DMI}(h'(X), \widetilde{Y}) \tag{30}$$

(2) Proposed new loss function $\mathcal{L}_{\mathbf{DMI}}$:
Definition:

$$\mathcal{L}_{\mathrm{DMI}}(Q_{h(X),\widetilde{Y}}) \triangleq -\log(\mathbf{DMI}(h(X), \widetilde{Y})) = -\log(|\det(\mathbf{Q}_{h(X),\widetilde{Y}})|) \tag{31}$$

(See how to calculate $\mathcal{L}_{\mathrm{DMI}}$ and Main Theorem in original paper. Look through the proof.)
My Question:
In the proof of Main Theorem, I think,

$$\mathbf{DMI}(h(X), \widetilde{Y}) = \mathbf{DMI}(h(X), Y)|\det(\mathbf{T}_{Y \to \widetilde{Y}})| \tag{32}$$

doesn't hold. Because, when using property (iv) to derive this result, the assumption $h(X) \perp \widetilde{Y}|Y$ may not hold. When training on classifier h on $\widetilde{Y}$, Y should be less informative for $h(X)$ than $\widetilde{Y}$.

References

[1] Ghosh, A., Kumar, H., and Sastry, P. Robust loss functions under label noise for deep neural networks, 2017.

[2] Wang, X., Kodirov, E., Hua, Y., and Robertson, N. M. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters.

[3] Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 322–330.

[4] Xu, Y., Cao, P., Kong, Y., and Wang, Y. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems* (2019), pp. 6222–6233.

[5] Zhang, Z., and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels, 2018.