Review of Beyond Standard Labels

Mohammed El mendili *November 2, 2020*

It is well known that the tremendous success that Deep Learning has been able to attend in the last decade is largely attributed to the use of **human-annotated** large datasets [3]. This is a big limitation for the field as labels come usually with an expensive cost. Moreover, noise/error in these annotations could easily break down the performance of Deep Neural Networks (DNNs) [1]. To alleviate these problems, Learning with Noisy Labels (LNL) and Semi-supervising Learning (SSL) were developed. In this review, we will begin by presenting a LNL approach in *Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels* [1], then a SSL approach *FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence*, and finally we will discuss how these two approaches could be used together to enhance DNNs performances in *DivideMix: Learning with Noisy Labels as Semi-supervised Learning* [3].

Mathematical Notation

We will be considering classification problems (K classes). Let's denote $f^\theta(x)$ (vector of size K) the softmax distribution output of a DNN (parametrized by θ), $D = \{(x_i, y_i), i \in (1, .., n)\}$ a clean labeled dataset, $D^\eta = \{(x_i, \hat{y_i}), i \in (1, .., n)\}$ a noisy dataset (driven by the noise η) and $U = \{u_i, i \in (1, .., \gamma n)\}$ ($\gamma \in]0, 1[$) an unlabeled dataset. We denote $R_L(f)$ and $R_L^\eta(f)$ the empirical risks associated to the loss L on D and D^η respectively.

1 Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels [1]

Due to their rich capacity, DNNs are easily prone to overfit to noisy labels [1]. Thus, (inevitable) errors in the labeling process can easily hamper their performance. Also, it is sometimes more beneficial to have many samples with noisy labels than few samples with correct ones (especially when examples are cheap but labels are expensive).
The paper [1] is motivated by recent works [4] that demonstrated, under some assumptions, that replacing the Cross-entropy loss (CCE) by the Mean Absolute Error (MAE) leads to a robust model against noise:

Theorem 1 *If the noise η is uniform and conditionally independent of inputs (i.e. $\forall i, \forall j, k \in [1, .., n], \mathbf{P}(\hat{y_i} = k|y_i = j, x) = \mathbf{P}(\hat{y_i} = k|y_i = j) = (1 - \eta)1_{j=k} + \frac{\eta}{K-1}1_{j\neq k}$ with $\eta < \frac{K-1}{K}$, and if the loss function L is symmetric (i.e. $\forall x, \forall f, \sum_{j=1}^K L(f^\theta, j) = C$, where C is a constant).*
Then, L is noise-tolerant (i.e. if f is a global minimizer for R_L then it is also a global minimizer for R_L^η)

The MAE loss verifies the above conditions for the uniform noise. Hence, it has been proposed as a noise-robust loss in [4]. However, authors in [1] showed that this comes at the expense of training time and accuracy. More specifically, the gradient of the loss has the following expression for CCE and MAE:

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

In CCE gradient, higher attention is paid, through the term $\frac{1}{f_\theta(x_i)}$, for samples with smaller $f_\theta(x_i)$ (they are the most challenging ones). This is good when training with clean data as it correctly directs and facilitate the training. The MAE gradient lacks this property as it considers all samples being equals which makes the training more difficult (slow convergence and decrease inaccuracy as showed in [1]) but, in the same time, robust to noisy samples.

A new class of loss functions

In order to leverage the power of the **implicit weighting scheme** of CCE and the **noise-robustness** of MAE, authors in [1] proposed the **Box-Cox Transformation** [5] as a loss function:

$$L_q(f(x), j) = \frac{(1 - f_j(x))^q}{q}, q \in]0, 1]$$

In this case the previous gradients have the following form:

$$\frac{\partial L(f^\theta(x_i), y_i)}{\partial \theta} = -f_{y_i}^\theta(x_i)^{q-1} \nabla_\theta f_{y_i}^\theta(x_i)$$

Since $q \leq 1$ the attention paid to difficult samples (with small $f_{y_i}^{\theta}$) is **less** than CCE and **more** than MAE. The parameter q somehow defines the trade-off between noise-robustness and the better learning dynamics. In fact, the limit $q = 0$ corresponds to CCE and $q = 1$ corresponds to MAE [1]. Since a tighter bound on $\sum_{j=1}^{K} L(f^{\theta}(x_j), y_j)$ imply more robustness to noise [1], authors in [1] suggested a truncated loss (for $f^{\theta}(x_j) \leq k$, the loss is considered as constant) and showed that the above mentioned bound is tighter. k is a hyper-parameter and depends on the noisiness of the data. However, this loss is difficult to train especially in the beginning of the training process when most of the softmax values are small (and hence the gradient is null in the truncated loss). To solve this optimization problem, authors in [1] used the *Alternative Convex Search* (ACS) [6] with pruning (Algorithm 1 in [1]). They also showed that, under different scenarios and datasets, the previous losses (L_q and truncated L_q) give good accuracy while being robust to noise.

2 FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence [2]

Semi-supervised Learning (SSL) provides a framework to improve a model's performance using unlabeled data (often cheap to obtain). Common SSL methods includes **Consistency regularization** and **Pseudo-labeling**:

- **Consistency Regularization:** Relies on the intuitive idea that a model should output the same results for different perturbations of the same input.
- **Pseudo-Labeling:** Relies on the idea that we should use the model's predictions to obtain artificial labels (usually "hard" labels, i.e. the argmax of the model's output distribution).

The loss proposed in FixMatch [2] leverages these two main ideas (Here, H designs the cross entropy function, α designs a **weak** augmentation of input and A designs a **strong** augmentation):

$$L = \underbrace{\frac{1}{n}\sum_{i=1}^{n} H(y_i, f_{y_i}^{\theta}(\alpha(x_i)))}_{\substack{\textbf{Standard cross entropy term on weakly augmented labeled samples (aligned with consistency regularization)}}} + \lambda_u \underbrace{\frac{1}{\gamma n}\sum_{i=1}^{\gamma n} 1(\max(f^{\theta}(u_i)) \geq \tau) H(\text{argmax}(f^{\theta}(\alpha(u_i))), f^{\theta}(A(u_i)))}_{\textbf{Cross-entropy term between confident } (\tau \textbf{ is a threshold}) \textbf{ artifical label generated from a weakly augmented input and the softmax distribution generated from a strongly augmented input using the DNN (aligned with pseudo-labeling)}}$$

where λ_u is the strength given to the unlabeled loss. Note that here the pseudo-labels are created from a weakly augmented inputs, but the loss is enforced against the model's prediction for a strongly augmented input. The authors in [2] showed that this gives improvements in the results. For the weak augmentation, they used a standard flip-and-shift augmentation strategy on images inputs (i.e. randomly with probability 50%, flip images horizontally and randomly translate them vertically and horizontally). For the strong augmentation, they based their experiments on RandAugment [7] and CTAAugment [8]. In [2], authors showed that even though FixMatch is comparatively simpler than other works in literature, it consistently outperforms state-of-the-art results on benchmarks and achieves good results even in the low-labeled regime. Also, they performed an ablation study that confirmed the utility of the model choices (augmentations, loss, thresholding, etc).

3 DivideMix:Learning with Noisy Labels as Semi-supervised Learning [3]

In [3], authors try to build a bridge between SSL and LNL by designing *DivideMix*, a model that learns noisy labels in a semi-supervised manner. The main idea of DivideMix is to discard labels that are most probably noise and use them to regularize the training in a SSL manner. DivideMix consists on training two **independent** DNNs in the same time and then use each one to teach the other on a batch level. Below the most important methods:
Co-Divide: DNNs tend to learn the clean samples faster than the noisy ones which leads to a higher loss for the latter [3]. Hence, DivideMix tries to fit, *for each DNN*, a two-components **Gaussian Mixture Model** (GMM) on the per-sample cross-entropy loss distribution:

$$\forall i, l_i = H\Big(\hat{y_i}, f^{\theta}(x_i)\Big) \sim N(\mu_1, {\sigma_1}^2) + N(\mu_2, {\sigma_2}^2), \mu_1 < \mu_2 \in R, \sigma_1, \sigma_2 > 0$$

where the parameters of the GMM are fitted using an **Expectation-Maximization** (EM) algorithm. Then, DivideMix divides the batch of the **other** DNN using a threshold τ on $w_i = P\Big(N(\mu_1, {\sigma_1}^2)|l_i\Big)$(Indeed, the distribution with smaller mean is the distribution of losses of clean samples). The reason why they use one DNN to split the batch of the other is to

avoid the so-called **confirmation bias** phenomenon (i.e. if it happens that the model learns an erroneous representation on the data, it will continue confirming its beliefs by classifiying contradictory labels as noise). Also, since the models needs to be *warmed-up* at the beginning of training (before fitting the GMM), an entropy penalty cost is added to the lost function to penalize confident predictions at the beginning of training. This helps to avoid a possible early overfitting to noisy labels.
Label-Refinement and Co-Guessing: Here, DivideMix uses MixMatch [9] (similar to FixMatch [2]). First, all the samples are augmented, then the labels of clean samples (obtained using the split described above) are adjusted (**label-refinement**) using the clean probability w_i and the models prediction (a mean guided by w_i), sharpened and mixed using the procedure of MixMatch [9]. For the unlabeled samples, their predictions are **co-guessed** using an average of the prediction of the two DNNs (over M augmented samples) (Algorithm 1 in [3]). Finally, the loss used to train the model is the weighted (strengths are hyper-parameters) sum of (1) Cross-entropy on the clean subset (2) Mean-Squared error loss between the model predictions and the average of its prediction on a set of augmented samples of the same input and (3) Regularization term related to a uniform prior over classes to regularize the model's output average on the batch.
Regarding the results, authors in [3] showed that DivideMix consistently provides substantial performance improvements compared to the state-of-the-art models.

4 Comments and Comparisons

- In [1,3], the validation of the models used pre-defined noise scenarios (Uniform noise, class-dependant noise, etc). This includes, implicitly, a background knowledge on the structure of the noise. Thus, a validation on a real-world noisy dataset would be more accurate to assess the performance of these methods.
- The ideas presented in [3] build a connection between LNL and SSL. This is very promising for next works as the two problems are conceptually correlated (samples with noisy labels could, in some extent, considered as unlabeled).
- In [2], FixMatch ourperformed MixMatch [9] in all the experiments that the authors performed. One can replace the MixMatch step performed in [3] by FixMatch to obtain, probably, better improvements on the performance.
- The set of loss function proposed in [1] is indeed more robust-to-noise than CCE. However, the optimization of this loss could be really challenging as it is known for MAE-like functions [1]. A more detailed research on the optimization method that fits best to this loss should be performed.
- In FixMatch [2], the choise of the threshold τ is crucial. In fact, choosing a high τ would be catastrophic in the case where the dataset contains very few labeled examples as the second term of the equation will be, in general, null in the first epochs (the model is less confident in the beginning of training) and then could overfit very quickly to the (few) labeled examples.
- The results presented in these methods are very promising. Leveraging SSL and LNL could be helpful to fields where gathering large amounts of data is challenging (e.g. medicine). However, all the tests were made on images and hence it may be more interesting to test on different types of inputs such as texts.

References

[1] Zhang and Sabuncu. , *"Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels", NeurIPS 2019.*

[2] Sohn et al. , *"FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence", arXiv 2020*

[3] Li et al. , *"DivideMix: Learning with Noisy Labels as Semi-supervised Learning", ICLR 2020*

[4] Ghosh et al., *"Robust loss functions under label noise for deep neural networks." In AAAI, pages 1919–1925, 2017.*

[5] George EP Box and David R Cox., *"An analysis of transformations." Journal of the Royal Statistical Society. Series B (Methodological), pages 211–252, 1964.*

[6] Bazaraa et al., *"Nonlinear programming: theory and algorithms." John Wiley Sons, 2013.*

[7] Ghosh et al., *"Making risk minimization tolerant to label noise." Neurocomputing, 160:93–107, 2015.*

[8] Azadi et al., *Auxiliary image regularization for deep cnns with noisy labels. arXiv preprint arXiv:1511.07069, 2015.*

[9] Berthelot et al., *"MixMatch: A Holistic Approach to Semi-Supervised Learning" arXiv:1905.02249v2 [cs.LG] 23 Oct 2019*