ADAPTIVE GENERALIZED CROSS-ENTROPY LOSS FOR SOUND EVENT CLASSIFICATION WITH NOISY LABELS

Jun Deng,1* Chunhui Gao,1 Qian Feng,1 Xinzhou Xu,2 Zhaopeng Chen1,3†

1Deep Learning Department, Agile Robots AG, Munich, 81477, Germany
jun.deng@agile-robots.com

2 School of Internet of Things, Nanjing University of Posts and Telecommunications, P. R. China
xinzhou.xu@njupt.edu.cn

3TAMS (Technical Aspects of Multimodal Systems), Department of Informatics, University of Hamburg

ABSTRACT

Considering the high cost of manually annotated large-scale datasets for superior sound event classifier performance, the data collection process has shifted to using the Internet, which facilitates easier user-contributed audio and metadata collection. However, label noise is inevitable. To address the problems caused by label noise, several types of noise-robust loss functions have been proposed recently as alternatives to the commonly categorical cross-entropy (CCE) loss, one of which is the generalized cross-entropy (GCE) loss, which demonstrates state-of-the-art performance. However, GCE cannot realize sufficient noise robustness and satisfactory accuracy simultaneously. Thus, we propose adaptive GCE loss, which automatically adapts to noisy labels in every batch to achieve adequate noise robustness and sufficient accuracy. We conducted experiments and found that the classification accuracy of the proposed loss demonstrated 4.7% and 1.2% absolute performance. However, GCE cannot realize sufficient noise robustness and satisfactory accuracy simultaneously. Thus, we propose adaptive GCE loss, which automatically adapts to noisy labels in every batch to achieve adequate noise robustness and sufficient accuracy. We conducted experiments and found that the classification accuracy of the proposed loss demonstrated 4.7% and 1.2% absolute improvement over the CCE and GCE baselines, respectively. We also demonstrate that clean data consumption in the proposed loss is dramatically reduced by more than 75% compared with CCE.

Index Terms— Sound event classification, label noise, cross-entropy, deep learning

1. INTRODUCTION

Sound event classification (SEC) research has attracted increasing attention in recent years. SEC facilitates a wide range of applications from context recognition to surveillance [1, 2, 3] and it is also useful for daily-life environment monitoring to improve user interactions with artificial intelligence systems [4, 5].

To develop sound event classifiers with effective performance, large-scale labeled datasets have been released recently with the development of Internet resources, e.g., Freesound [6], AudioSet [7], and FSDKaggle2018 [8]; however, the labelings of these datasets are noisy. For example, in AudioSet, the rate of label error rate is estimated to be greater than 50% for approximately 17.7% of classes1, and in FSDKaggle2018, at least 65% of the annotations are non-verified in each category [8]. Empirical evidence indicates that deep networks are robust to some amount of label noise [9, 10]; however, significant label noise can introduce many problems and challenges, e.g., performance reduction, increased complexity of learned models, and changes in learning requirements [11].

The research progress of SEC with label noise is still lacking behind compared with related works in the computer vision field [12, 13]. In recent years, in light of the progress in improving noise robustness in computer vision, some methods have germinated, especially in terms of model-based modification and learning strategy [14, 15]. Soft bootstrapping [16] handles noisy labeling by updating the prediction objective by combining of the noisy label and the current prediction. Batch-wise loss masking [17] prevents data with corrupted labels from negative impact to the total loss by discarding the loss values of it. The recently proposed generalized cross-entropy (GCE) loss [18] can be considered as the generalization of the CCE loss [19, 20] and mean absolute error (MAE). Ionescu et al. [21] provided an empirical evaluation of these loss functions in the SEC context [21]. Among these loss functions, the GCE loss demonstrated outstanding performance in an experiment [21].

Nevertheless, the GCE loss cannot perform perfectly due to the hyperparameter in the loss, which typically involves a tuning procedure that results in a trade-off between noise robustness and accuracy. To address this issue, we propose an adaptive GCE loss function by utilizing an adaptive mechanism in every batch rather than tuning the hyperparameter to achieve sufficient noise robustness for noisy labeling data and the high accuracy for clean data. The proposed approach realizes other benefits, e.g., reduced computational workload for hyperparameter tuning and insensitivity to other hyperparameters.

Our primary contributions are summarized as follows. First, we propose an adaptive GCE loss and report a thorough evaluation of the proposed loss compared with i.e., categorical cross-entropy (CCE) loss and GCE loss. Compared with these baseline methods, the proposed adaptive GCE loss demonstrates significant improvement in terms of the classification accuracy. In addition, the experimental results also revealed that the proposed loss can outperform the baseline CCE system with less data consumption.

2. NOISE ROBUST LOSS FUNCTIONS

The essence of training a deep neural network (DNN) is to update the network weights to minimize a loss function that describes the discrepancy between the predictions from the network and the ground-truth labels. However, when labels are corrupt (referred to label

1First two authors contributed equally.

This research has received funding from the German Research Foundation (DFG) and the National Science Foundation of China (NSFC) in project Crossmodal Learning, DFG TRR-169/NSFC6162136606, partially supported by European projects H2020 STEP2DYNA (691154) and ULTRACEPT (778602), as well as National Science Foundation of China under Grant No. 61801241

1http://www.eduardofonseca.net/FSDDenoisy18k/
noise), updating weights can be suboptimal [21], which hinders model convergence, and can suppress the model’s performance. Noise robust loss functions can be helpful in such cases. Here, we briefly review the GCE loss and present the proposed adaptive GCE loss.

2.1. Generalized cross-entropy loss: $L_q$ loss

Ghosh et al. [22] proved and empirically demonstrated that mean absolute error (MAE) is robust against noisy labels; however in [18], the author argued that MAE is inappropriate for DNNs with challenging datasets because it results in significantly slow convergence and a substantial reduction in test accuracy. To make use of the benefits of the noise robustness provided by MAE and the implicit weighting scheme of CCE, the $L_q$ loss function, which applies a negative Box-Cox transformation, has been proposed:

$$L_q = \frac{1}{q} - \left( \frac{\sum_{k=1}^{K} y_k b_k}{q} \right)^q,$$

where $q \in (0, 1)$ is treated as a hyperparameter [18]. When $q$ approaches 0, the loss function is considered to be CCE, and when $q = 1$, the loss becomes MAE. Clean data and noisy data simultaneously exist in a user-contributed dataset. When trained with the GCE loss on these clean data, increasing $q$ slows down the convergence rate and reduces the classification accuracy, similar to how MAE behaves. However, a system with a small $q$ value is likely vulnerable to noisy data, which results in poor performance. In such a case, a SEC system should achieve sufficient accuracy and noise robustness for both clean and noisy data.

2.2. Adaptive generalized cross entropy loss: Adaptive $L_q$

As discussed previously, the selection of $q$ in GCE results in a trade-off between noise robustness and sufficient accuracy. However, if we assume that exponent $q$ is a variable dependent on a metric representing the degree of noise in the data, then $q$ can be adjusted automatically according to the degree of error-annotated data. Inspired by the soft nearest neighbor (SNN) concept [23], we can use it as this metric, which reflects how close pairs of representations from the same class are, relative to pairs of representations from different classes. Intuitively, we can conceptualize this metric by imagining that data points with noisy labels are obviously distant from other data points of the same class relative to data points with correct labels. The $SNN$ of a given batch with $b$ samples $(x_i, y_i)$ can be computed as follows [24]:

$$SNN = \frac{\sum_{i=1}^{b} \sum_{j=1}^{b} e^{-\frac{\|x_i - x_j\|^2}{T}}}{\sum_{i=1}^{b} \sum_{j=1}^{b} e^{-\frac{\|x_i - y_j\|^2}{T^2}}},$$

where $x$ and $y$ represent the input vector and the label of a given sample respectively. The temperature, $T$, is to control the relative importance given to the distances between pairs of points. For example, at low temperatures, $SNN$ is dominated by the small distances and the actual distances between widely separated representations are almost irrelevant; when the temperature is high, the distances between widely separated points can influence the $SNN$. To achieve the best performance of $T$, we treated it with a hyperparameter and experimented. From Equation (2), we can see that

Figure 1: Data distributions of training set, noisy subset and clean subset. (a) Training set split into noisy and clean data. (b) Further noisy data split into extra small, small, medium and large subsets. (c) Clean data split into small, medium, and large subsets.

the $SNN$ is negatively correlated to the ratio of the sum of distances between the sample $(x_i, y_i)$ and other intra-class samples to the sum of distances between the sample $(x_i, y_i)$ and all the other samples in the batch.

In our situation, however, we expect $q$ to be positively correlated to the ratio of the sum of the distances between intra-class and inter-classes. Here we modified Equation (2) to be positively correlated to the sum of the distances between the target data and the other data from the same class:

$$qa = 1 - SNN,$$

which can be considered an adaptive parameter $q$ (we refer to this as $qa$). We then input this into Equation (1) to formalize the adaptive GCE loss as follows:

$$L_{qa} = \frac{1}{q} - \left( \frac{\sum_{k=1}^{K} y_k b_k}{q} \right)^{qa},$$

It is worth noting that unlike GCE manually tune $q$, our proposed method is to adaptively selecting $q$ based on $SNN$ on every batch, but still needs to tune $T$.

3. EXPERIMENTS

Here, we discuss the experiments conducted to compare the proposed noise-robust adaptive $L_q$ loss function with the CCE and GCE baseline systems and to observe the performance when the amount of clean data was reduced.

3.1. Dataset

We used the FSDnoisy18k [6, 21] dataset in our experiments. The FSDnoisy18k contains 18,532 mono audio clips across 20 sound

2http://www.eduardofonseca.net/FSDnoisy18k/
Table 1: Training Data. The training dataset is split into several subsets with different amounts of noisy and clean clips, e.g., in the “small noisy + clean” subset, there is a tiny portion of noisy clips (1,003) and clean clips (1,772).

<table>
<thead>
<tr>
<th>Subsets</th>
<th>Noisy clips</th>
<th>Clean Clips</th>
</tr>
</thead>
<tbody>
<tr>
<td>extra small noisy + clean</td>
<td>500</td>
<td>1,772</td>
</tr>
<tr>
<td>small noisy + clean</td>
<td>1,003</td>
<td>1,772</td>
</tr>
<tr>
<td>medium noisy + clean</td>
<td>5,000</td>
<td>1,772</td>
</tr>
<tr>
<td>large noisy + small clean</td>
<td>10,000</td>
<td>443</td>
</tr>
<tr>
<td>large noisy + medium clean</td>
<td>10,000</td>
<td>886</td>
</tr>
<tr>
<td>large noisy + large clean</td>
<td>10,000</td>
<td>1,329</td>
</tr>
<tr>
<td>large noisy + clean</td>
<td>10,000</td>
<td>1,772</td>
</tr>
<tr>
<td>all</td>
<td>15,813</td>
<td>1,772</td>
</tr>
</tbody>
</table>

Figure 2: Convolutional neural network (CNN) with three convolutional layers and one dense layer [25].

Figure 3: Average accuracy (%) of running the proposed loss model 12 times, trained with different T using “all” train set.

Table 2: Average classification accuracy (%) and 95% confidence interval (10 runs) obtained by several losses on four subsets with different amounts of noisy clips, all clean clips and one complete set with all training. (best accuracy is shown in bold)

<table>
<thead>
<tr>
<th>Subsets</th>
<th>CCE loss</th>
<th>( L_q )</th>
<th>Adaptive ( L_{qa} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>extra small noisy + clean</td>
<td>60.4 ± 0.6</td>
<td>63.0 ± 0.3</td>
<td>63.7 ± 0.8</td>
</tr>
<tr>
<td>small noisy + clean</td>
<td>63.2 ± 1.0</td>
<td>65.4 ± 0.8</td>
<td>64.9 ± 1.7</td>
</tr>
<tr>
<td>medium noisy + clean</td>
<td>66.9 ± 1.0</td>
<td>70.7 ± 0.7</td>
<td>71.3 ± 1.0</td>
</tr>
<tr>
<td>large noisy + clean</td>
<td>68.5 ± 0.5</td>
<td>73.3 ± 0.6</td>
<td>74.5 ± 0.7</td>
</tr>
<tr>
<td>all</td>
<td>70.6 ± 0.6</td>
<td>74.1 ± 0.5</td>
<td>75.3 ± 0.5</td>
</tr>
</tbody>
</table>

3.2. Baseline system

All audio clips were transformed to 96-band, log-mel spectrogram as the input representation [21]. Figure 2 presents a three-layer CNN following a previously reported architecture [21, 25].

The learning strategy configuration was the same as that reported in the literature [21]. Here, the mini-batch size was 64, and the Adam optimizer was used with an initial learning rate of 0.001, which was reduced if no improvement was observed for a patience of five epochs. Early stopping was also applied to terminate training when the validation accuracy stopped improving for a patience of 15 epochs.

3.3. Results and discussion

Before evaluating the performances of the loss functions, we experimented to find the best parameter \( T \) of adaptive CCE loss. At first we trained the model using different values of \( T \), and we observed that when \( T \leq 100 \), the model cannot converge every time. Therefore, we studied \( T \) ranging from \( 10^2 \) to \( 10^6 \). We observed that the highest accuracy is achieved when the value of \( T \) is 1,000 (Figure 3). Thus, we always set \( T = 1,000 \) in our loss function in later experiments.

The experimental results obtained on five data subsets of different sizes with the different loss functions are summarized in Table 2 and Figure 4. To evaluate the proposed adaptive \( L_{qa} \)'s performance, we used accuracy that is a quintessential classification metric for our situation where classification problems are well class balanced and not skewed. Moreover, this metric has been used in considerable classification problems [25, 26, 27].

As presented in Table 2, for all subsets (from top to bottom), the average accuracy (of 12 runs with 95% confidence interval) of the three different losses improves with an increasing amount of noisy data. From left to right, the proposed adaptive \( L_{qa} \) loss significantly outperforms the baselines on different subsets, with the exception of the small noisy group, where \( L_q \) demonstrates better
performance. Specifically, when trained on a large noisy subset, the proposed adaptive $L_{qa}$ increased the accuracy by 6.0% and 1.2% over the CCE and GCE baselines, respectively. When training the adaptive $L_{qa}$ on the entire set, the accuracy was increased by 4.7% and 1.2% over the baselines. These results indicate that the proposed adaptive $L_{qa}$ works effectively with noisy labels.

In addition, we discovered that adaptive $L_{qa}$ requires less data expense than CCE and $L_q$ to achieve equal performance. For example, we observed that adaptive $L_{qa}$ performs better on a large noisy (74.5% accuracy) subset than $L_q$ on all training set (74.1% accuracy), which can be beneficial in cases where insufficient amounts of training data are available.

Figure 4 presents a boxplot [28] of the classification accuracy (%) distribution obtained by the three losses on five subsets with increasing amounts of noisy clips, including information about central tendency and the mean, median, and distribution of the data. As can be seen, the proposed adaptive $L_{qa}$ loss consistently outperforms the $L_q$ loss in terms of the median. Compared with the other losses, the “whiskers” of CCE on different subsets (except for the large noisy subset) are much longer, which means they vary more widely (representing instability) in accuracy. In contrast, $L_q$ and adaptive $L_{qa}$ tend to center more on the average accuracy (representing stability). In terms of skew (representing data asymmetry), the whiskers of the proposed loss on the small noisy subset, medium noisy subset, large noisy subset, and the complete training set are pretty even on either side of the median and mean, which means that the classification accuracy of the proposed loss on these subsets was distributed evenly.

The results obtained with different amounts of clean clips are outlined in Table 3. As shown in the second to the bottom rows, with the amount of clean data on the subsets varying from small portion (221 clips) to all (1,772 clips), all the performances with different losses actually provide a boost of 5.1% at least over the benchmark with CCE on totally clean set with the help of using a portion of noisy data for replacement of the missing part of the clean data and the accuracy increase is enlarged to 14.9% when training with adaptive $L_{qa}$ on large noisy and clean subset. In addition, training on the large noisy and small clean subsets is 1% better than CCE on the large noisy and total clean subsets. This suggests that the proposed adaptive $L_{qa}$ can save 75% of clean subsets consumption relative to the traditional CCE loss function.

4. CONCLUSIONS

In this paper, we have proposed a noise-robust loss function that is adaptive based on label noise estimation at the batch-level. The proposed loss function can achieve the adequate noise robustness for noisy labeling data and sufficient accuracy for clean data. The experimental results indicated that the proposed adaptive GCE achieved 4.7% and 1.2% absolute improvement compared with the CCE loss and GCE loss baseline systems, respectively, in terms of accuracy. In addition, compared with traditional CCE loss, training with the proposed adaptive GCE on a large noisy subset and a small portion of clean data can save 75% of expense of clean subset, which requires much less labelling effort. In the future, we plan to exploit to use high level feature embeddings for $S/N/N$, for example, a pre-trained model used as a feature extractor to compute high level representations to feed the $S/N/N$ equation [29, 30].

5. REFERENCES


---

3 All performance differences discussed in this paper are expressed in terms of absolute accuracy.


