Neural Networks 139 (2021) 358–370

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

A noisy label and negative sample robust loss function for DNN-based distant supervised relation extraction

Lihui Deng, Bo Yang *, Zhongfeng Kang, Shantian Yang, Shihu Wu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

ARTICLE INFO

Article history:
Received 7 November 2020
Received in revised form 8 March 2021
Accepted 19 March 2021
Available online 1 April 2021

Keywords:
Loss function
Noisy label learning
Class imbalance
Distant supervised relation extraction
Gradient analysis

ABSTRACT

As a major method for relation extraction, distantly supervised relation extraction (DSRE) suffered from the noisy label problem and class imbalance problem (these two problems are also common for many other NLP tasks, e.g., text classification). However, there seems no existing research in DSRE or other NLP tasks that can simultaneously solve both problems, which is a significant insufficiency in related researches. In this paper, we propose a loss function which is robust to noisy label and efficient for the imbalanced class dataset. More specific, first we quantify the negative impacts of the noisy label and class imbalance problems. And then we construct a loss function that can minimize these negative impacts through a linear programming method. As far as we know, this seems to be the first attempt to address the noisy label problem and class imbalance problem simultaneously. We evaluated the constructed loss function on the distantly labeled dataset, our artificially noised dataset, human-annotated dataset of Docred, as well as the artificially noised dataset of CoNLL 2003. Experimental results indicate that a DNN model adopting the constructed loss function can outperform other models that adopt the state-of-the-art noisy label robust or negative sample robust loss functions.

1. Introduction

Relation extraction aims to recognize the semantic relations between entity pairs from the plain text and has been used in many natural language processing (NLP) tasks such as knowledge graph construction (Fu & Ma, 2019; Geng et al., 2020; Takanobu et al., 2019; Vashishth et al., 2018; Xu & Barbosa, 2019; Zhang et al., 2019; Zheng et al., 2017), question answering (Geng et al., 2020; Qu et al., 2018; Zhang et al., 2019; Zhou et al., 2018), etc. In recent years, the deep neural network (DNN) based relation extraction model has been proposed and become popular (Li, Long et al., 2020; Lin et al., 2016; Zeng et al., 2017, 2015; Zhang et al., 2019; Zheng et al., 2017). DNN based relation extraction model generally requires a large amount of labeled data. Therefore, distantly supervised relation extraction (DSRE) has been proposed to reduce human labor in labeling the data by automatically labeling the text through the existing knowledge base (Li, Long et al., 2020; Luo et al., 2017; Riedel et al., 2010; Zeng et al., 2017, 2015). To date, most DSRE tasks are based on DNN models.

However, two major obstacles exist in the DSRE task: the noisy label problem and the class imbalance problem.

The noisy label problem (i.e., a large number of labels are incorrect) can severely impede the DNN model's performance (Han et al., 2018; Li, Long et al., 2020; Lin et al., 2016; Ren et al., 2017; Sun et al., 2019; Wu et al., 2019). For example, in the commonly used NYT-10 dataset (Riedel et al., 2010), up to 35% of the relation facts are wrongly labeled (Li, Long et al., 2020), and in the recently published Docred datasets (Yao et al., 2019), the ratio of wrongly labeled data exceeds 41.4%. Due to the noisy label problem in the DSRE task, the performance of the DNN models is often seriously affected. Existing researches in DSRE address the noisy label problem by adopting multi-instance learning strategy (Li, Long et al., 2020; Luo et al., 2017; Qu et al., 2018; Wu et al., 2019). However, this strategy can only be adopted to multi-sentence scenario, which is a draw back for learning intra-class relation. Therefore, inspired by other deep learning tasks (Amid et al., 2019; Ghosh et al., 2017; Wang et al., 2019; Zhang & Sabuncu, 2018), in this paper we adopted a modified loss function to address the noisy label problem. Generally, there are two types of strategies in adopting modified loss functions. One is to penalize the model for assigning no probability to labels other than the given label. This strategy adds extra terms in the loss function that involve the predicted probability of labels other than the given label, preventing the model from overconfident about the given label. Nevertheless, it is difficult to determine the probability assigned to each class, especially when the labels' quantity is significantly unbalanced. As a result, this strategy is

* Corresponding author.
E-mail addresses: denglh2019@126.com (L. Deng), yangbo@uestc.edu.cn (B. Yang), kangzhf@gmail.com (Z. Kang), yangshantian2009@gmail.com (S. Yang), wushihu1998@gmail.com (S. Wu).

https://doi.org/10.1016/j.neunet.2021.03.030
0893-6080/

not suitable for the DSRE task since the labels in DSRE are long-tail distributed (Han et al., 2018; Yao et al., 2019). The other strategy preventing model overfitting to noisy labels is using bounded-value loss function (i.e., the value of the loss function is bounded), under which the empirical risk gap (i.e., the empirical risk gap between the DNN model trained from the noisy data and the DNN model trained from the clean data) can be bounded. Minimizing this bound can close the gap between the DNN model trained from noisy data and the DNN model trained from clean data, thus, alleviate the negative impact from noisy label problem. Popular loss functions adopting this strategy include bounded mean square error (BMSE) (Ghosh et al., 2017), generalized cross entropy (GCE) (Zhang & Sabuncu, 2018), polynomial loss function (Gong et al., 2019). In this paper, our research focuses on the bounded-value loss function strategy to address the noisy label problem in the DSRE task. Other than the label noise, external disturbance may also exist in the sequential text (e.g., outliers, ambiguous words), and can affect the convergence and stability of the trained model. Researchers developed methods to filter the noise in sequential system (Dong et al., 2020; Stojanovic et al., 2020; Stojanovic & Prsic, 2020; Tao et al., 2020; Zhou et al., 2020). However, since the DSRE task contains rich sequential text, the influence of the noise in the sequential text is limited, therefore is not considered in this paper.

The other severe problem for DSRE task is the class imbalance problem. In fact, in the practical scenario of DSRE, most of the entity pairs do not express a semantic meaning and are negative samples (i.e., the samples with *negative labels* that do not express any semantic meaning). For example, in the NYT-10 dataset, 70% of the entity pairs are negative samples (Ye & Luo, 2019). In the distantly supervised dataset of Docred (Yao et al., 2019), we find that the DSRE task needs to extract 1,508,320 relation facts from 39,124,970 entity pairs, which means over 96% of the entity pairs are negative samples. Hence the majority of training data are negative samples, which will impede the DNN model's prediction accuracy for positive samples (i.e., the samples with *positive labels* that express semantic meaning) (Wen et al., 2016; X. Zhang et al., 2017; Zhang et al., 2020). Existing researches address this negative impact in their fields by adopting loss functions robust to negative samples to increase the prediction accuracy for positive samples. Popular loss functions include range loss (X. Zhang et al., 2017), focal loss (FL) (Lin et al., 2017), and class-balanced loss (CBL) (Cui et al., 2019). Adopting these loss functions can accelerate the learning process for positive samples. However, these loss functions are not robust to the noisy label. Therefore, the DNN models trained with these loss functions are prone to noisy samples.

The noisy label problem and negative sample problem also commonly existed in other NLP tasks such as named entity recognition (NER) (Li, Sun et al., 2020; Namysl et al., 2020), text classification (Chen et al., 2019; Marcelino et al., 2018), machine reading comprehension (Dasigi et al., 2019; Liu et al., 2020). Therefore, in this paper we take DSRE as example, and propose a loss function that is robust to noisy labels and negative samples to address these two problems simultaneously. To achieve this, firstly we need to quantify the negative impact from the noisy label problem and class imbalance problem, then design a loss function that can minimize these negative impacts. For the noisy label problem, we adopt the bounded-value loss function and then propose the calculation method for its empirical risk gap bound, which is used to represent the negative impact from the noisy label problem. For the class imbalance problem, we first analyze the learning process of the DNN model and find that the negative samples' learning process will significantly interfere the positive samples. Prediction accuracy of the positive samples will tend to decrease in the initial learning stage. Then, we propose the turning value for the positive samples, which can guarantee that the prediction accuracy for positive samples will tend to increase when reached this value. Therefore, the loss function with higher turning value can be more robust to negative impact from negative samples. In this paper, the turning value is used to evaluate the negative impact from class imbalance problem.

After quantifying the negative impact from the two problems, we construct a loss function with minimized empirical risk gap bound and maximized turning value to address the noisy label problem and class imbalance problem. This construction process is a multi-target programming task from mathematical perspective. To simplify this task, the construction is through a linear programming task that targets minimizing the empirical risk gap bound and constrained the turning value for positive samples higher than a threshold. The workflow for this paper is shown in Fig. 1. When compared to existing noisy label robust or negative sample robust loss function, our loss function can achieve the minimal empirical risk gap while guarantee the convergence rate at initial learning stage, thus can achieve better performance against noisy label and negative sample.

In summary, the contributions of this paper are as follows.

(1) We find that positive samples' learning process will be interfered by negative samples in the DSRE task, and the prediction accuracy for positive samples will tend to decrease in the initial learning stage. To address this negative impact, we propose the turning value for positive samples. By increasing the turning value of the loss function, positive samples' learning process can be more robust to the negative samples.
(2) The negative impact from noisy label problem and class imbalance problem is quantified in this paper. Based on these quantified negative impact, we propose the construction method for noisy label robust and negative sample robust loss function. Compared to existing loss functions that are noisy label robust or negative sample robust, our loss function is more robust to label noise and can guarantee the model's convergence rate at initial learning stage.
(3) As far as we know, this is the first attempt to address the noisy label problem and class imbalance problem simultaneously in DSRE task or other NLP tasks. And a loss function is constructed to achieve this. The model's performance adopting our constructed loss function is evaluated on the distantly labeled dataset, artificially noised dataset, and human-annotated dataset of Docred. The model adopts the constructed loss function has achieved higher performance than other state-of-the-art noisy label robust or negative sample robust loss functions, respectively. We also evaluated our loss function on the named entity recognition (NER) task, and conducted experiments on the artificially noised dataset of CoNLL 2003. The experimental results in CoNLL 2003 show that our loss function also yields the best performance compared with existing loss functions.

The structure of this paper is as follows. Section 2 provides the notations for DSRE task and the preliminaries for Section 3. The quantification method for noisy label problem and class imbalance problem, and the construction method for our loss function are contained in Section 3. Section 4 presents and analyzes the experiments and results. Section 5 concludes the paper.

2. Preliminaries

In this section, we provide the notations for DSRE task and the gradient analysis for the DNN model.

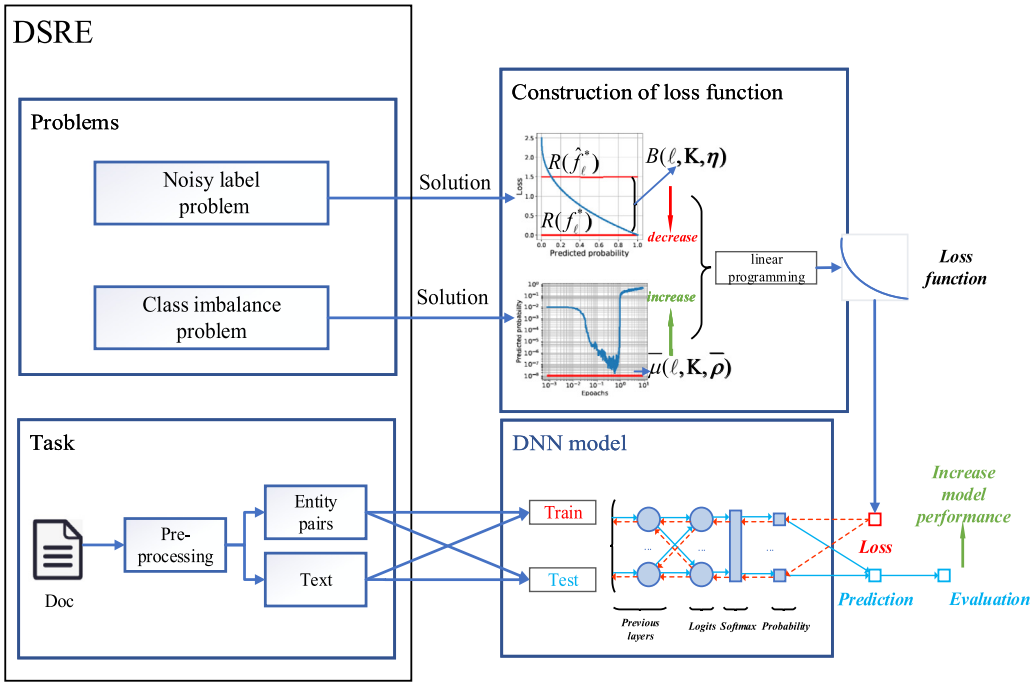


Fig. 1. Workflow for adopting the constructed loss function in DSRE task, where $R(\bar{f}_\ell^*)$ and $R(f_\ell^*)$ can represent the performance of model learned from noisy data and clean data, and $B(l, K, \eta)$ is the empirical risk gap bound, $\bar{\mu}(l, K, \bar{\rho})$ is the average turning value for positive labels.

2.1. Notations and definitions for distantly supervised relation extraction

Relation extraction: Given an input text and its entities, the task of relation extraction is to extract the relation facts between the entity pairs. For a dataset with N_e entity pairs contain possible relational facts, the data is $\{(\mathbf{x}, y)_i\}_{i=1}^{N_e} \in (\mathcal{X} \times \mathcal{Y})^{N_e}$, where $(\mathbf{x}, y)_i$ is the ith data sample, the input data $\mathbf{x}$ is the numerical representation of entity pair and its related text, $\mathcal{X}$ is the domain for $\mathbf{x}$. y is the ground true label of $\mathbf{x}$, and $\mathcal{Y} = \{0, 1, \ldots, K\}$, where K is the number of categories for positive relations, $\mathcal{D}$ is the clean distribution of the input data $(\mathbf{x}, y)$. In this paper, the negative label is represented with $y = 0$, and the other scenarios are positive labels.

Let $f(\mathbf{x}, \theta)$ be the probability of each category predicted by the DNN model, where $f : \mathcal{X} \to \mathbb{R}^{K+1}$, and θ is the model's parameter. When given sample $(\mathbf{x}, y)$, we denote $f_y(\mathbf{x}, \theta)$ be the prediction accuracy for this sample. The misclassification rate on $\mathcal{D}$ is measured by the empirical risk under 0–1 loss:

$$R_{0-1}(f) = \mathbb{E}_{\mathcal{D}}[\ell_{0-1}(f(\mathbf{x}, \theta), y)]$$

where

$$\ell_{0-1}(f(\mathbf{x}, \theta), y) = \begin{cases} 1 & , \quad \underset{j \in \mathcal{Y}}{argmax}\, f_j(\mathbf{x}, \theta) \neq y, \\ 0 & , \quad else. \end{cases}$$

The objective for training process is to minimize $R_{0-1}(f)$. According to Bartlett et al. (2006), ℓ is said to be classification-calibrated if given a clean dataset, the minimizer of $R_\ell(f)$ can minimize $R_{0-1}(f)$ as well. However, ℓ_{0-1} is hard to be optimized in practice, therefore a surrogate loss function that is differential and convex, and is classification-calibrated (e.g., cross entropy) $\ell(f(\mathbf{x}, \theta), y)$ is generally adopted.

Noisy label problem: The label from the knowledge base has high probability of being incorrect (Riedel et al., 2010; Yao et al., 2019), therefore the relation labels labeled by knowledge base are regarded as noisy label $\tilde{y}$. The label noise can be assumed to be uniform or class conditional.

Uniform noise:

$$\mathbb{P}[\tilde{y} \neq y|y] = \eta$$

where η is the noise rate from uniform noise.

Class conditional noise:

$$\mathbb{P}[\tilde{y} = j|y] = \eta_{yj}, j \neq y$$

where $\tilde{y}$ is the category index of the noisy label, η_{yj} is the probability of labeled j when given the input data $\mathbf{x}$.

The empirical risk under ℓ for model trained from noisy data is $R_\ell^\eta(f) = \mathbb{E}_{\mathcal{D}^\eta}[\ell(f(\mathbf{x}, \theta), \tilde{y})]$. Let f_ℓ^* and $\hat{f}_\ell^*$ be the global minimizer of $R_\ell(f)$ and $R_\ell^\eta(f)$. Due to the negative impact from noisy label, there exists a gap between $R_\ell(\hat{f}_\ell^*)$ and $R_\ell(f_\ell^*)$. If the loss value is bounded, then the bound of the gap can be calculated and minimized to make the model robust to noisy label.

Class imbalance problem: In this paper, the labels that express semantic meaning are defined as positive labels; otherwise, they are negative labels. And in the DSRE task, we mainly concern about the prediction accuracy for positive labels. However, the label's quantity in the DSRE task is highly unbalanced. For example, the labels' quantity in the distantly labeled dataset of Docred are long-tail distributed, as shown in Fig. 2. Therefore, in the dataset distribution $\mathcal{D}$, the probability for model to encounter negative labels $\mathbb{P}[y = 0]$ are significantly higher than positive labels $\mathbb{P}[y = j]$ for $j \neq 0$. Therefore, the model's learning process for positive samples will be impeded by negative samples (Wen et al., 2016; X. Zhang et al., 2017; Zhang et al., 2020). The analysis of how negative samples interfere the positive samples is conducted through gradient analysis. The notations for gradient analysis are provided in the next subsection.

2.2. Notations for gradient of the DNN model

The notations for the gradient of the DNN model are provided in this subsection. And since the softmax layer is widely used to transform the logits output of DNN models into predicted

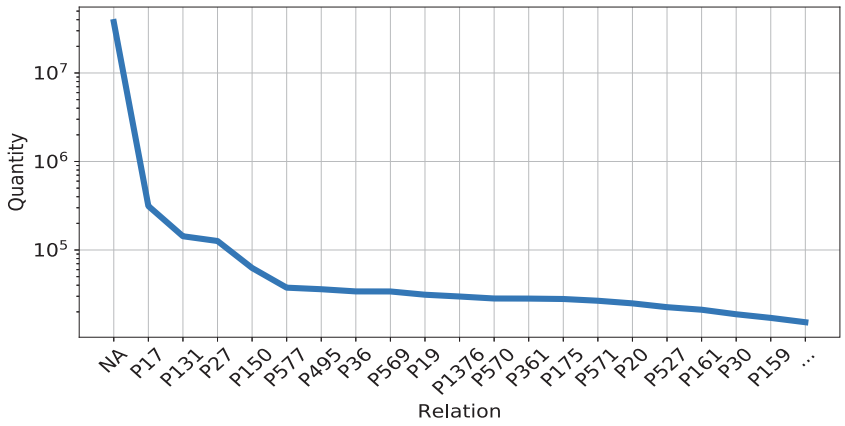


Fig. 2. Labels' quantity for the most commonly occurred relations in the distantly labeled dataset in Docred.

probability, in this paper, the gradient is mainly studied at the softmax layer. The form of softmax layer is:

$$f_j(\mathbf{x},\theta)=\frac{\exp(\mathbf{u}_j(\mathbf{x},\theta))}{\sum_{k=0}^{K}\exp(\mathbf{u}_k(\mathbf{x},\theta))}$$

where $f_j(\mathbf{x},\theta)$ is the predicted probability for label j, and we define $f_y(\mathbf{x},\theta)$ as the prediction accuracy for sample $(\mathbf{x},y)$. $\mathbf{u}$ is the logits output of the neural network, given the label y, the derivatives of $f_y(\mathbf{x},\theta)$ to $\mathbf{u}(\mathbf{x},\theta)$ is:

$$\frac{\partial f_y(\mathbf{x},\theta)}{\partial \mathbf{u}_j(\mathbf{x},\theta)}=\begin{cases} f_y(\mathbf{x},\theta)(1-f_y(\mathbf{x},\theta)) & ,j=y,\\ -f_y(\mathbf{x},\theta)f_j(\mathbf{x},\theta) & ,j\neq y.\end{cases} \tag{1}$$

When loss value is only related to $f_y(\mathbf{x},\theta)$, the loss function ℓ can be seen as a function of $f_y(\mathbf{x},\theta)$:

$$l(f_y(\mathbf{x},\theta))=\ell(f(\mathbf{x},\theta),y) \tag{2}$$

Let $\frac{\partial l(f_y(\mathbf{x},\theta))}{\partial f_y(\mathbf{x},\theta)}=l'(f_y(\mathbf{x},\theta))$, then the gradient at the softmax layer is:

$$\frac{\partial l(f_y(\mathbf{x},\theta))}{\partial \mathbf{u}_j(\mathbf{x},\theta)}=\begin{cases} l'(f_y(\mathbf{x},\theta))f_y(\mathbf{x},\theta)(1-f_y(\mathbf{x},\theta)) & ,j=y,\\ -l'(f_y(\mathbf{x},\theta))f_y(\mathbf{x},\theta)f_j(\mathbf{x},\theta) & ,j\neq y.\end{cases} \tag{3}$$

Then the training process of model is to update θ through its gradient, the gradient of θ from sample $(\mathbf{x},y)$ is:

$$\nabla\mathbf{f}=\frac{\partial l(f_y(\mathbf{x},\theta))}{\partial \mathbf{u}(\mathbf{x},\theta)}\frac{\partial \mathbf{u}(\mathbf{x},\theta)}{\partial\theta} \tag{4}$$

Then we denote the gradient update from sample $(\mathbf{x},y)$ associated with label j:

$$\nabla\mathbf{f}_j=\frac{\partial l(f_y(\mathbf{x},\theta))}{\partial \mathbf{u}_j(\mathbf{x},\theta)}\frac{\partial \mathbf{u}_j(\mathbf{x},\theta)}{\partial\theta} \tag{5}$$

A simple diagram of back-propagation is shown in Fig. 3, where the red lines are the gradient associated with label j.

Batch learning is widely applied in DNN model training, in this paper we assume θ is updated through gradient from batch of samples. Denote $\mathbf{G}_j(\mathcal{D},\theta)$ be the gradient associated with label j from a batch of samples, then the expectation of $\mathbf{G}_j(\mathcal{D},\theta)$ is:

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}[\mathbf{G}_j(\mathcal{D},\theta)]\\ &=N\Big(\mathbb{P}[y=j]\mathbb{E}_{\mathbf{x}|y=j}[\frac{\partial l(f_j(\mathbf{x},\theta))}{\partial \mathbf{u}_j(\mathbf{x},\theta)}\frac{\partial \mathbf{u}_j(\mathbf{x},\theta)}{\partial\theta}]\\ &\quad+\sum_{k\neq j}\mathbb{P}[y=k]\mathbb{E}_{\mathbf{x}|y=k}[\frac{\partial l(f_k(\mathbf{x},\theta))}{\partial \mathbf{u}_j(\mathbf{x},\theta)}\frac{\partial \mathbf{u}_j(\mathbf{x},\theta)}{\partial\theta}]\Big)\end{aligned} \tag{6}$$

where N is the batch size, $\mathcal{D}$ is the distribution for sample $(\mathbf{x},y)$.

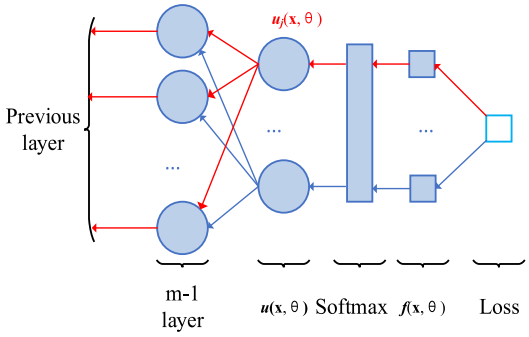


Fig. 3. Diagram of gradient update for deep network using softmax layer before output layer. (The red line is the gradient update associated with label j). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. The proposed loss function

Our loss function is constructed in this section. In Section 3.1, the empirical risk gap bound is proposed to alleviate the noisy label problem. In Section 3.2, we conduct gradient analysis to find out the cause for the negative impact from the class imbalance problem. In Section 3.3, the turning value for positive samples' prediction accuracy is proposed, and numerical experiments for the turning value are conducted in Section 3.4. In Section 3.5, We propose the construction method for the loss function. In Section 3.6, we provide the implementation details for our loss function.

3.1. The empirical risk gap bound

In this subsection, the bound for $R_\ell(\hat{f}_\ell^*)-R_\ell(f_\ell^*)$ under different noise type is proposed in Theorems 1 and 2, respectively.

First, to calculate the bound for $R_\ell(\hat{f}_\ell^*)-R_\ell(f_\ell^*)$, the bound for $\sum_{k=0}^{K}l(f_k(\mathbf{x},\theta))$ is given in Lemma 1.

Lemma 1. *If $l''(p)\geq 0$ and $l'(p)\leq 0$ when $p\in(0,1)$, then the upper bound for $\sum_{k=0}^{K}l(f_k(\mathbf{x},\theta))$ is $Kl(0)+l(1)$, and the lower bound for $\sum_{k=0}^{K}l(f_k(\mathbf{x},\theta))$ is $(K+1)l(\frac{1}{K+1})$.*

The proof of Lemma 1 can be found in Appendix A. Then the bound of $R_\ell(\hat{f}_\ell^*)-R_\ell(f_\ell^*)$ under uniform noise is given in Theorem 1.

Theorem 1. *Under uniform noise, if* $\eta < \frac{K}{K+1}$, $l''(p) \geq 0$, $l'(p) \leq 0$ *when* $p \in (0, 1)$, *then* $R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*)$ *is bounded by:*

$$0 \leq R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*) \leq \frac{\eta(Kl(0) + l(1) - (K+1)l(\frac{1}{K+1}))}{K - \eta(K+1)} \tag{7}$$

The proof of Theorem 1 can be found in Appendix B. Additionally, the bound of $R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*)$ under class conditional noise is given by Theorem 2.

Theorem 2. *Under class conditional noise, if* $l(1) = 0$ *and* $R_\ell(f_\ell^*) = 0$, *and* $l''(p) \geq 0$, $l'(p) \leq 0$ *when* $p \in (0, 1)$, *then* $R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*)$ *is bounded by:*

$$0 \leq R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*) \leq \frac{l(0)\mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}]}{min_{y=0,1,\ldots,K}(1 - \sum_{j\neq y}\eta_{yj})} \tag{8}$$

The proof of Theorem 2 can be found in Appendix C.

After the bound for $R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*)$ is proposed, minimizing this bound is able to close the gap between the DNN model trained from noisy data and the DNN model trained from clean data. And can alleviate the negative impact from noisy label to the trained model. In this paper, we conduct a linear programming task that target at minimizing the bound for $R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*)$ to construct the noisy label robust loss function. For the sake of simplicity, $R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*)$ is assumed to be under uniform noise. And we define $B(l, K, \eta)$ be the upper bound of $R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*)$:

$$B(l, K, \eta) = \frac{\eta(Kl(0) + l(1) - (K+1)l(\frac{1}{K+1}))}{K - \eta(K+1)} \tag{9}$$

Through minimizing $B(l, K, \eta)$, model trained with loss function l is more robust to noisy labels. And since minimizing $Kl(0) + l(1) - (K+1)l(\frac{1}{K+1})$ is identical to minimizing $\frac{\eta(Kl(0)+l(1)-(K+1)l(\frac{1}{K+1}))}{K-\eta(K+1)}$, therefore we define $b(l, K)$ as our minimization target:

$$b(l, K) = Kl(0) + l(1) - (K+1)l(\frac{1}{K+1}) \tag{10}$$

Besides the empirical risk gap bound, gradient also plays an essential role in controlling the DNN model's learning process and is discussed in the next subsection.

3.2. Gradient analysis for the class imbalance problem

In this subsection, we analyze the negative impact of class imbalance problem through gradient analysis. After learning the sample $(\mathbf{x}, y)$, the predicted probability for y would be increased, and the predicted probability for other labels will be decreased. The gradient for label j obtained by $(\mathbf{x}, y)$ are

$$\nabla \mathbf{f}_j = \frac{\partial l(f_y(\mathbf{x}, \theta))}{\partial \mathbf{u}_j(\mathbf{x}, \theta)} \frac{\partial \mathbf{u}_j(\mathbf{x}, \theta)}{\partial \theta}$$

When $j = y$, $\frac{\partial l(f_y(\mathbf{x},\theta))}{\partial \mathbf{u}_j(\mathbf{x},\theta)} < 0$, updating the gradient $\nabla \mathbf{f}_j$ will increase model's predicted probability for label j when input $\mathbf{x}$ (Rumelhart et al., 1986). In contrast, when $j \neq y$, then $\frac{\partial l(f_y(\mathbf{x},\theta))}{\partial \mathbf{u}_j(\mathbf{x},\theta)} > 0$, then the predicted probability for label j will decrease.

However, as depicted in Fig. 4(a), when the model is initialized, all samples' feature representation is randomly dispersed. The model cannot discriminate between positive and negative samples. Since the gradient of DNN models only related to the sample's feature representation and label, the learning process of samples with closely related feature distribution will interfere with each other. When learning a batch of samples, for sample $(\mathbf{x}, y)$, its learning process could be interfered by other samples in this batch. Since the feature distribution of different labels is closely distributed in the initialized DNN model, we can ignore the difference of $\frac{\partial \mathbf{u}_j(\mathbf{x},\theta)}{\partial \theta}$ between different input $\mathbf{x}$, then the form of Eq. (6) can be transformed as:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\mathbf{G}_j(\mathcal{D}, \theta)] \\
&= N \Big(\mathbb{P}[y = j]\mathbb{E}_{\mathbf{x}|y=j}[\frac{\partial l(f_j(\mathbf{x}, \theta))}{\partial \mathbf{u}_j(\mathbf{x}, \theta)}] \\
&\quad + \sum_{k\neq j} \mathbb{P}[y = k]\mathbb{E}_{\mathbf{x}|y=k}[\frac{\partial l(f_k(\mathbf{x}, \theta))}{\partial \mathbf{u}_j(\mathbf{x}, \theta)}] \Big) \mathbb{E}_{\mathcal{D}}[\frac{\partial \mathbf{u}_j(\mathbf{x}, \theta)}{\partial \theta}]
\end{aligned} \tag{11}$$

Denote the term in Eq. (11) be error term $\mathbf{g}(\mathcal{D}, \theta)$:

$$\begin{aligned}
&\mathbf{g}_j(\mathcal{D}, \theta) \\
&= N \Big(\mathbb{P}[y = j]\mathbb{E}_{\mathbf{x}|y=j}[\frac{\partial l(f_j(\mathbf{x}, \theta))}{\partial \mathbf{u}_j(\mathbf{x}, \theta)}] \\
&\quad + \sum_{k\neq j} \mathbb{P}[y = k]\mathbb{E}_{\mathbf{x}|y=k}[\frac{\partial l(f_k(\mathbf{x}, \theta))}{\partial \mathbf{u}_j(\mathbf{x}, \theta)}] \Big)
\end{aligned} \tag{12}$$

Then if $\mathbf{g}_j(\mathcal{D}, \theta) > 0$, according to Rumelhart et al. (1986), the prediction accuracy for label j will be decreased even if the sample with label j has been learned in this batch.

Based on the knowledge above, Proposition 1 is proposed to show that labels with low quantity will tend to decrease in the beginning of training.

Proposition 1. *For the randomly initialized DNN model with the softmax layer before outputs, if the derivative of its loss function* $l'(p) < 0$ *when* $p \in (0, 1)$, *then the model's prediction accuracy for samples with label j will tend to decrease at the beginning of training if* $\mathbb{P}[y = j] < \frac{1}{K+1}$.

Please see Appendix D for the proof of Proposition 1. For the DSRE task, since negative samples are the majority in the training dataset, the positive samples' learning process will be interfered by negative samples as long as their feature distribution are close. As shown in Fig. 4(b), the model still lacks the discriminative ability for commonly occurred positive samples even after learning 101 873 documents for four epochs. Therefore, when training is complete, the model's prediction accuracy for positive samples will still be impeded. In the next subsection, we propose the turning value to alleviate the decrease of positive samples' prediction accuracy.

3.3. The turning value for positive samples

In this subsection, we propose the turning value for the prediction accuracy of positive samples. First, an assumption is made as follows.

Assumption 1. If the DNN model is randomly initialized, and has been trained for T batches, and $\mathbf{g}_j(\mathcal{D}, \theta^t) > 0$ for $t = 1, 2, \ldots, T$, then the model will not learn information from samples with label j. Thus the difference between input $\mathbf{x}$ for the expectation of $\frac{\partial \mathbf{u}_j(\mathbf{x},\theta)}{\partial \theta}$ and $f_j(\mathbf{x}, \theta)$ can be ignored.

Then the turning value is proposed in Proposition 2.

Proposition 2. *Under Assumption 1 and the same conditions from Proposition 1, for label j with* $\mathbb{P}[y = j] < \frac{1}{K+1}$, *if the adopted loss function satisfies that:* $l'(0) \to -\infty$, $l''(p) \geq 0$, $l'(p) \leq 0$, *and* $(l'(p)p)' \leq 0$ *when* $p \in (0, 1)$, *then there exists turning value* $\mu_j \in (0, \frac{1}{K+1})$, *which is the solution for* $\mathbb{P}[y = j]l'(\mu_j) - l'(1) = 0$. *When* $\mathbb{E}_{\mathbf{x}|y=j}[f_j(\mathbf{x}, \theta^t)] \leq \mu_j$, *the prediction accuracy for label j will tend to be increased.*

Please see Appendix E for the proof of Proposition 2. Denote $\rho = \{\mathbb{P}[y = 0], \mathbb{P}[y = 1], \ldots, \mathbb{P}[y = K]\}$ be the ratio of the

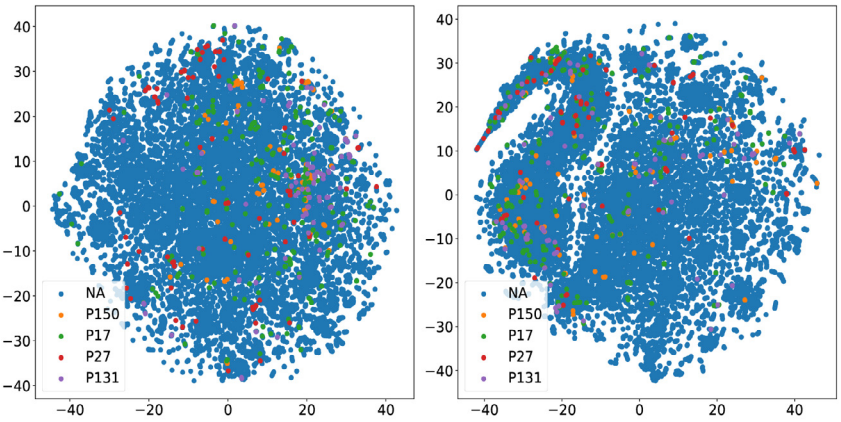


Fig. 4. The DNN model's feature distribution (conducted by t-SNE van der Maaten & Hinton, 2008) for commonly occurred positive samples and negative samples during the training process. The training data is from Docred(distantly supervised), which contains 101 873 documents (The detailed information of Docred will be given in Experiments Section).

labels' quantity, and let $\bar{\rho}$ be the median value for ρ, $\bar{\mu}$ be the solution for:

$$\bar{\rho} l'(\bar{\mu}) - l'(1) = 0 \tag{13}$$

In this paper, we assume $\bar{\rho} \leq \frac{1}{K+1}$, and $\bar{\rho}$ is set to $\frac{1}{K+1}$ if the median value for ρ is higher than $\frac{1}{K+1}$. In this paper, the negative impact from negative samples will be represented by $\bar{\mu}(l, K, \bar{\rho})$. Increasing $\bar{\mu}(l, K, \bar{\rho})$ can accelerate the model's learning process for positive samples, therefore will be the important part in the construction of our loss function.

3.4. Numerical experiments for the turning value

Numerical experiments are conducted to evaluate whether the turning value can bound the prediction accuracy for positive samples. The experiments are conducted on the distantly labeled dataset of Docred (Yao et al., 2019) with BiLSTM model, and we adopt the GCE loss function with $\bar{\mu} = [10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}]$, then the average prediction accuracy for positive samples along the training process are shown in Fig. 5. As depicted in Fig. 5, the average prediction accuracy for positive sample will first tend to decrease to $\bar{\mu}$. Therefore, $\bar{\mu}$ can be used to represent the negative impact from negative samples, and higher $\bar{\mu}$ can increase the robustness of model to negative samples.

3.5. Construction of the loss function

Based on the empirical risk gap bound $B(l, K, \eta)$ and the average turning value $\bar{\mu}(l, K, \bar{\rho})$, the construction method for our loss function will be proposed in this subsection. The form of the constructed loss function $l_C(f_y(\mathbf{x}))$ is:

$$l_C(f_y(\mathbf{x})) = \sum_{n=0}^{M} \mathbf{a}_n f_y(\mathbf{x})^{\frac{n}{M}} \tag{14}$$

where M is a positive constant, in order to satisfy the condition $l'(0) \to -\infty$ in Proposition 2, we set $M \geq 3$. $\mathbf{a} \in \mathbb{R}^{M+1}$ is the parameter for the term of the constructed loss function. To address the noisy label problem and the class imbalance problem, $b(l_C, K)$ need to be minimized and $\bar{\mu}(l_C, K, \bar{\rho})$ need to be maximized. To simplify this multi-target programming problem, in this paper we construct our loss function through linear programming which target at minimizing $b(l_C, K)$, and set constraint to make $\bar{\mu}(l_C, K, \bar{\rho})$ higher than a threshold.

Generally, the constraints for our constructed loss function are:

(a) $\bar{\mu}(l_C, K, \bar{\rho}) \geq C_1$.
(b) $l'_C(p) \leq 0$ when $p \in (0, 1)$.
(c) $l''_C(p) \geq 0$ when $p \in (0, 1)$.
(d) $(l'(p)p)' \leq 0$ when $p \in (0, 1)$.
(e) $\|\mathbf{a}\|_\infty \leq C_2$
(f) $l_C(0) = C_3$.
(g) $l_C(1) = 0$.

where C_1, C_2, C_3 are constants. Constraint (a) is to reduce the negative impact from negative samples, and $\bar{\mu}$ is the solution for $\bar{\rho} l'(\bar{\mu}) - l'(1) = 0$. When $l''(p) \geq 0$, constraint $\bar{\mu}(l_C, K, \bar{\rho}) \geq C_1$ can be converted to $\bar{\rho} l'(C_1) \leq l'(1)$. Constraint (b), (c) is the necessary condition to keep the loss function convex and monotone decreasing on interval (0,1). Constraint (d) is to satisfy the necessary condition for Proposition 2. To simplify the constraints (b), (c), (d), we set d be a positive constant and $d \in (1, M)$, and $a_n < 0, n = 1, \ldots, d, a_m > 0, m = d+1, \ldots, M$, then we propose Lemma 2 to convert constraints (b), (c), (d) to linear constraints.

Lemma 2. *Let $D(p) = \sum_{n=1}^{d} A_n p^{F(n)} - \sum_{m=d+1}^{M} A_m p^{F(m)}$, where $F(n)$ is a monotone increasing function for n, if $A_n \leq 0, n = 1, \ldots, M$, and $D(1) = \sum_{n=1}^{d} A_n - \sum_{m=d+1}^{M} A_m \leq 0$, then $D(p) \leq 0$ when $p \in (0, 1)$.*

The proof for Lemma 2 is shown in Appendix F. According to Lemma 2, if $a_n < 0, n = 1, \ldots, d$ and $a_n > 0, n = d+1, \ldots, M$, constraints (b), (c), (d) can be converted to $l'_C(1) \leq 0$, $-l''_C(1) \leq 0$, $l''(1) + l'(1) \leq 0$, respectively. Constraint (e) is to reduce the computation time. Constraints (f), (g) are the constraints of magnitude for the loss function.

When d is determined, the construction of $l_C(f_y(\mathbf{x}))$ can be converted to a linear programming problem that minimize $b(l_C, K)$ with the constraints (a)–(g):

$$\min_{\mathbf{a}} -\Big(\sum_{n=1}^{M} (K+1)^{(1-\frac{n}{M})} \mathbf{a}_n + \mathbf{a}_0\Big) \tag{15}$$

$$s.t. \sum_{n=1}^{M} \frac{n}{M} (C_1^{\frac{n-M}{M}} \bar{\rho} - 1) \mathbf{a}_n \leq 0 \tag{15a}$$

$$\sum_{n=1}^{M} \frac{n}{M} \mathbf{a}_n \leq 0 \tag{15b}$$

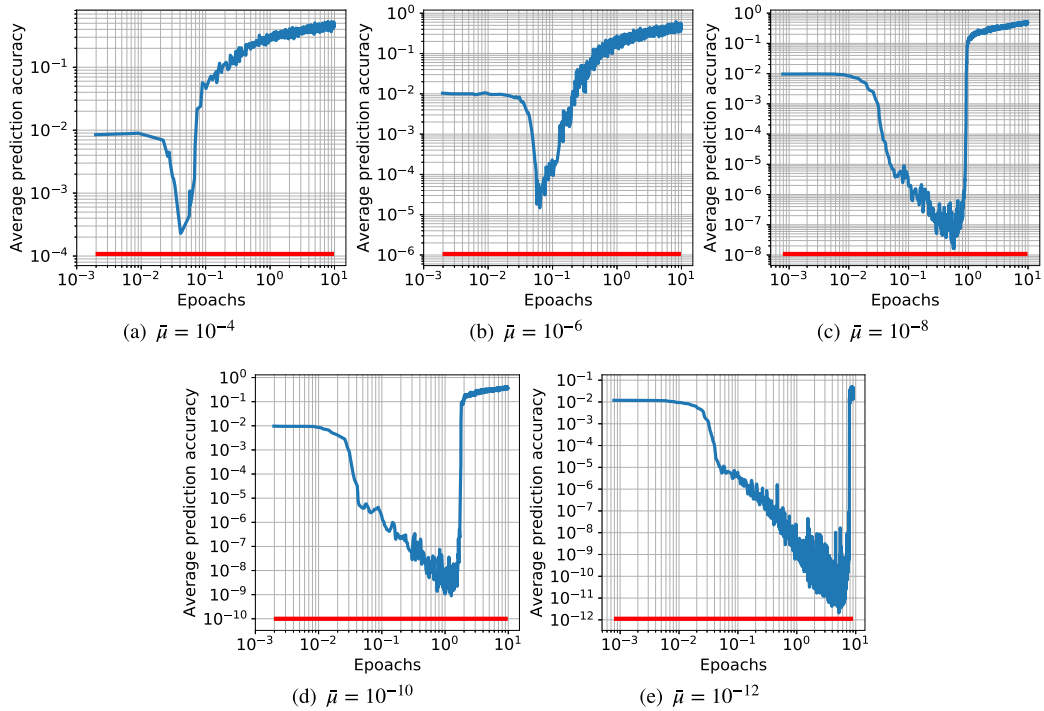


Fig. 5. Average predicted probability for positive samples along the training process(blue line), the red line represents the value for $\bar{\mu}$, $\bar{\mu}$ for subfigure (a), (b), (c), (d), (e) are 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12} respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\sum_{n=1}^{M} \frac{-n(n-M)}{M^2}\mathbf{a}_n \leq 0 \tag{15c}$$

$$\sum_{n=1}^{M} (\frac{n}{M} + \frac{n(n-M)}{M^2})\mathbf{a}_n \leq 0 \tag{15d}$$

$$\|\mathbf{a}\|_\infty \leq C_2 \tag{15e}$$

$$\sum_{n=0}^{M} \mathbf{a}_n = 0 \tag{15f}$$

$$\mathbf{a}_0 = C_3 \tag{15g}$$

$$\mathbf{a}_n < 0, n = 1, \ldots, d \tag{15h}$$

$$\mathbf{a}_m > 0, m = d+1, \ldots, M \tag{15i}$$

where Eq. (15) is the target for the linear programming task. Eq. (15a) is from constrain (a). Eq. (15b) is from constrain (b). Eqs. (15c), (15d), (15e) are from constraints (c), (d), (e), respectively. Eqs. (15f), (15g) are from constraints (f), (g), respectively. Eqs. (15h), (15i) are the necessary constraints in converting the construction process to a linear programming problem. The solution method are described in the next subsection.

3.6. Implementation detail

When d is determined, Eq. (15) can be minimized through linear programming algorithm. Thus, we set $d = 1, \ldots, M-1$, and solve the linear programming problem iteratively. The solution with minimal $b(l, K)$ is adopted as the optimized solution. The method we use to solve the linear programming problem is interior-point (Karmarkar, 1984) algorithm, which is applied through scipy[1] library. The pseudo code for the solution method is shown in Algorithm 1. For the detailed interior-point algorithm in solving the linear programming problem, please refer to Vaidya (1989). According to Vaidya (1989), the computational complexity of the interior-point algorithm is $O(M^{2.5}L)$, where L is the number of bits of variables. Thus the computational complexity of Algorithm 1 is $O(M^{3.5}L)$.

Algorithm 1 Construction method for our loss function

Input: Number of relation categories $K \in \mathbb{Z}_+$; median value of the labels' quantity ratio $\bar{\rho} \in (0, \frac{1}{K+1})$; hyper-parameters for the construction of loss function: $C_1 \in (0, \frac{1}{K+1})$; $C_2 \in [0, \infty)$, $C_3 \in [0, \infty)$, $M \geq 3$, $M \in \mathbb{Z}_+$.
Output: $\mathbf{a}^*$.

```
Set b* = ∞
for d = 1; d < M − 1; d++ do
    Solving Eq. (15) using interior-point algorithm.
    if b(l, K) < b* then
        b* ← b(l, K)
        a* ← a
    end if
end for
return a*
```

In this paper, we set $C_3 = 5$ to constrain the magnitude of the loss function. And we discover that C_2 and M will not influence the optimization outcome significantly, thus can be set as large as possible. Without loss of generality, we set $C_2 = Inf$ and $M = 100$. When set $M = 100$, $C_2 = Inf$, Algorithm 1 can be solved within a seconds with Intel(R) Core i7-7800X CPU, thus the computational burden of Algorithm 1 is acceptable. As for C_1, we can set C_1 near to $\frac{\bar{\rho}}{10}$ empirically, then find the best C_1 through grid search on the experimental results of the dataset.

4. Experiments

This section includes the experimental results on DSRE task and NER task. The experiments for DSRE task are based on Docred

[1] https://www.scipy.org/.

dataset (Yao et al., 2019), which contains 101 873 distantly labeled documents and 5053 human-annotated documents, where the human-annotated documents are divided as 3053 documents for training (train-annotated), 1000 documents for model validation (Dev), 1000 documents for model performance evaluation (Test). The category number for relation $K = 96$. The experiments for DSRE task are conducted on distantly labeled dataset, our artificially noised dataset, and human-annotated dataset.

The widely used metrics F1 score and AUC score are applied for performance evaluation. Considering that many entity pairs in Dev and Test dataset also existed in the training dataset, F1 score and AUC score for entity pairs that are not contained in the training dataset are also used to evaluate the prediction performance, and are defined as Ign F1 score and Ign AUC score (Yao et al., 2019).

Distantly labeled dataset: For distantly labeled dataset, assuming that the relation may exist between every entity pairs, the quantity for negative labels $Num_0 = 37,619,332$, the sum of the positive labels quantity $\sum_{k\neq 0} Num_k = 1,505,638$, thus the quantity of negative label is higher than the sum of the positive labels. ρ are estimated through $\rho_j = \frac{Num_j}{\sum_{k=0}^{K} Num_k}$, where Num_j is the quantity of label j in the dataset, in Docred, median value for labels' quantity ratio $\bar{\rho} = 1.67e{-}4$. According to Yao et al. (2019), the wrongly labeled rates for relation instance exist in inner-sentence and inter-sentence are 41.4% and 61.8%, respectively, while 46.4% relation instance associated with more than one sentence, therefore when assuming the noise to be uniform, we set the probability of wrong labeling: $\eta = 0.464 \times 0.618 + 0.536 \times 0.414 = 0.509$.

Artificially noised dataset: In this paper, the artificially noised dataset is generated by modifying the human-annotated training data. The noise type includes uniform noise and class conditional noise. For uniform noised data, the label of positive samples will be mapped to other random labels with probability of noise rate η. Denote n_{pn} be the number for labels that are mapped from positive labels to the negative label, then n_{pn} negative labels will be mapped to random positive labels. For class conditional noise, we first label the entity pairs in the human-annotated dataset through wikidata knowledge base, then estimate the wrongly labeling probability of mapping positive label y to other label j :

$$\eta_{yj} = \frac{I_{yj}}{\sum_{k\neq y} I_{yk}} \eta$$

where η is the parameter that determine the general noise rate, I_{yj} is the times that positive label y is labeled to j through the wikidata knowledge base. After mapping the positive label, n_{pn} negative labels are mapped to positive label with probability: $\eta_{1j} = \frac{I_{1j}}{\sum_{k\neq 1} I_{1k}}, j \neq 1$.

4.1. Experimental setup

The model applied in relation extraction task is BiLSTM from Yao et al. (2019). All models are trained with ADAM optimizer, with learning rate of 2e−4, and $\beta_1 = 0.9$, $\beta_2 = 0.999$. The other training procedures and model architectures for all experiments are identical, only the loss functions are changed.

4.2. Experiments on distantly labeled dataset

In this subsection, we first evaluate the performance of our loss function with different set of C_1 and C_2, then compare our loss function with other state of the art loss functions. Since $\frac{\bar{\rho}}{10} = 1.67e-5$, We test C_1 in $[1e-4, 1e-5, 1e-6, 1e-7]$, and set $C_2 = Inf$, $C_3 = 5$, $M = 100$, then evaluate our loss function on the distantly labeled dataset. Fig. 6(a) depicts that lower C_1 (e.g.,1e−7/1e−6) will tend to decrease the model performance in the initial steps, and the convergence steps will become slower. And when C_1 is higher (e.g., 1e−4/1e−5), the difference of performance between $C_1 = 1e-4$ and $C_1 = 1e-5$ is not significant, but higher C_1 can provide higher convergence rate. For this reason, a relatively large C_1 will benefit the convergence rate on the dataset.

As for C_2, different set of C_2 do not influence our constructed loss function. And as shown in Fig. 6(b), when set $C_1 = 1e-4$, $C_3 = 5$, $M = 100$ and test C_2 in $[5, 10, 20, Inf]$, different set of C_2 do not have significant influence on the model performance or convergence rate, therefore C_2 can be set as large as possible, and is set as Inf in this paper.

Therefore, we set $C_1 = 1e-4$, $C_2 = Inf$, $C_3 = 5$, $M = 100$, and compared the results with CE, three recently proposed bounded-value loss function: (1) GCE (Zhang & Sabuncu, 2018): Generalization of MAE and CE; (2) polynomial (Gong et al., 2019): Weighting the samples through its learning difficulty; (3) BMSE (Ghosh et al., 2017): Bound the loss value for mean square error; three loss function that is robust to negative sample: (1) FL (Lin et al., 2017): Loss function that increase the weight of hard samples. (2) CBL (Cui et al., 2019): Balance the weight of each class through its effective numbers. (3) Range loss (X. Zhang et al., 2017): Minimize the range of each class within one batch to learn long-tail dataset. The models' performance is reported in Table 1. Table 1 shows that our constructed loss function improved the model performance when compared with cross entropy and other loss functions that is robust to noisy label or negative sample.

Table 1
Evaluation score for Dev and Test dataset for model trained in Distantly labeled dataset with CE, Polynomial, GCE, BMSE, FL, CBL, Range loss, our constructed loss function, the best results are high-lighted in bold.

		CE	Noisy label robust			Negative sample robust			Ours
			Poly	GCE	BMSE	FL	CBL	Range	
Dev	F1	48.90	49.55	49.86	49.05	49.53	49.61	49.32	**51.85**
	Ign F1	43.89	45.27	45.56	43.45	44.43	44.68	44.07	**47.24**
	AUC	46.92	47.56	47.13	46.23	47.21	47.04	46.98	**50.46**
	Ign AUC	42.57	43.14	42.81	41.97	43.05	42.95	42.57	**46.24**
Test	F1	48.84	49.75	49.82	49.17	49.82	49.82	49.56	**51.81**
	Ign F1	44.17	45.03	45.47	43.23	44.38	44.56	44.16	**46.94**
	AUC	46.88	47.39	47.04	46.69	47.03	47.22	47.05	**50.23**
	Ign AUC	42.66	43.01	42.94	41.76	43.02	42.87	42.61	**46.13**

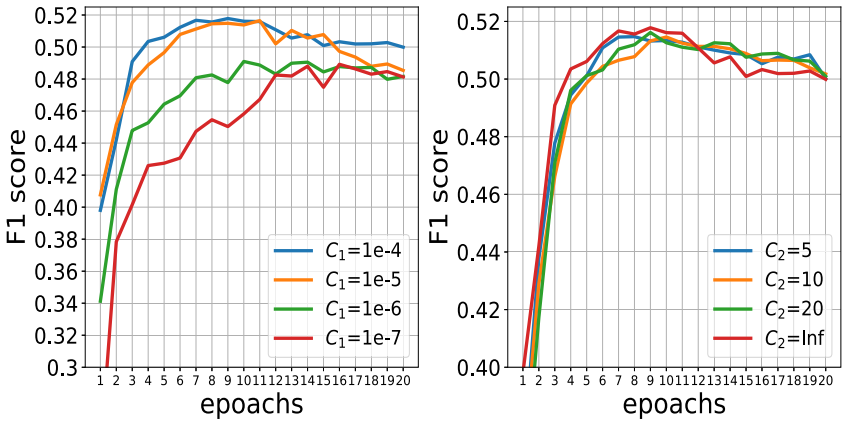
Table 1 shows that our constructed loss function improved the model performance when compared with cross entropy and other loss functions that is robust to noisy label or negative sample.

The complete learning procedures of constructed loss function and other loss on distantly labeled dataset are illustrated in Fig. 7. As shown in Figs. 7(a) and 7(b), our constructed loss function both achieved the best performance. As shown in Fig. 7(a), when compared with other noisy label robust loss functions, model adopting our loss function can achieve higher better performance in the first few epochs, and therefore outperform other models with existing noisy label loss functions. And when compared with other negative sample robust loss functions, since model adopting our loss function can be robust to noisy label, thus can also outperform models with existing negative sample robust loss functions few epochs after the training.

On the other hand, since the models performance on the Dev and Test are relatively close, we will evaluate the models' performance under different loss functions on the Dev dataset in the latter part of this paper.

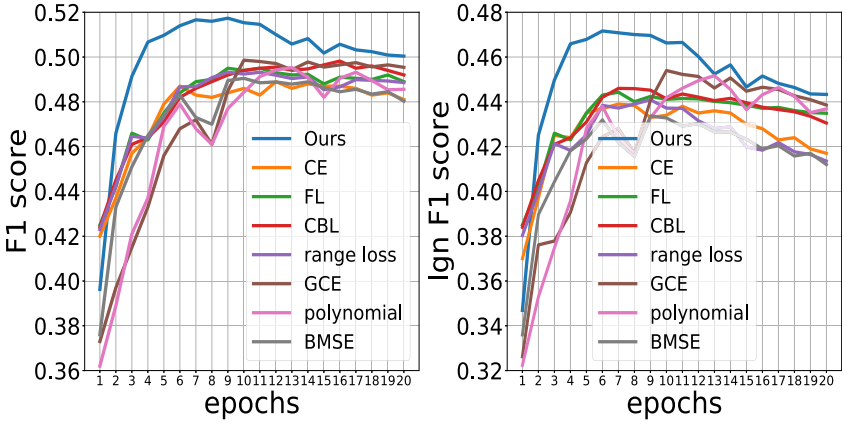
4.3. Experiments on artificially noised dataset

To evaluate the performance of our constructed loss function on different noise rate and noise type, experiments are conducted on the artificially noised dataset of Docred. We noised



(a) different set of C_1 (b) different set of C_2

Fig. 6. F1 score on Dev dataset against number of epochs for training with constructed loss function with different set of C_1 and C_2.



(a) F1 score (b) Ign F1 score

Fig. 7. F1 score(a) and Ign F1 score(b) on Dev dataset against number of epochs for training with different loss functions.

the human-annotated dataset with uniform noise at noise rate of [0.2,0.4,0.6,0.8] and with class conditional noise at noise rate of [0.2,0.4,0.6]. Considering that the human-annotated dataset's size is significantly smaller than distantly labeled dataset, we set the training epochs to 100. Other settings are identical to the distantly labeled dataset.

The result with artificially noised data is summarized in Table 2. For uniform noise and class conditional noise, the constructed loss function has achieved better performance, especially when the noise rate is higher.

4.4. Experiments on human-annotated dataset

We also evaluate our constructed loss function in the clean dataset. As depicted in Table 3, the performance of our loss function is slightly better than other negative sample robust loss functions on three of four evaluation scores, which prove the negative sample robustness of our loss function.

4.5. Experiments on named entity recognition

NER task also faced with noisy label problem and negative sample problem (Li, Sun et al., 2020; Namysl et al., 2020). Therefore we also conduct experiments on this task to evaluate the generalization of our loss function. Experiments are conducted on the widely used CoNLL 2003 dataset, which contains 170k

Table 2
F1 score and Ign F1 score for Dev dataset for model trained in artificially noised dataset with CE, Polynomial, GCE, BMSE, FL, CBL, Range loss, our constructed loss function, the best results are high-lighted in bold.

	Loss functions		Uniform noise				Class conditional noise		
			Noise rate				Noise rate		
			0.2	0.4	0.6	0.8	0.2	0.4	0.6
	CE	F1	49.08	47.11	43.18	35.11	48.37	42.55	30.86
		Ign F1	44.44	42.52	38.37	30.44	43.79	38.03	26.41
Noisy label robust	Poly	F1	49.81	48.37	44.90	36.95	48.74	44.51	33.07
		Ign F1	44.92	44.01	40.36	31.72	44.30	40.11	28.26
	GCE	F1	48.81	47.41	42.66	35.84	48.14	42.60	31.17
		Ign F1	44.39	42.62	38.35	30.85	43.84	38.45	26.97
	BMSE	F1	49.21	47.37	43.76	35.94	48.35	43.02	31.63
		Ign F1	44.23	42.44	38.15	30.98	43.45	38.07	27.26
Negative sample robust	FL	F1	49.21	47.33	43.45	35.12	48.28	42.77	30.89
		Ign F1	44.61	42.54	38.41	30.66	43.77	38.15	26.67
	CBL	F1	49.44	47.31	43.38	35.31	48.57	42.75	31.06
		Ign F1	44.55	42.58	38.73	30.81	43.90	38.11	26.54
	Range	F1	49.15	47.13	43.32	35.25	48.66	43.98	31.54
		Ign F1	44.33	42.18	38.16	30.11	43.13	37.67	26.11
	Ours	F1	**50.47**	**49.28**	**46.62**	**39.88**	**49.19**	**46.35**	**36.74**
		Ign F1	**45.57**	**44.79**	**41.72**	**35.66**	**44.80**	**42.03**	**31.82**

Table 3
F1 score and Ign F1 score for Dev dataset for model trained in human-annotated dataset with CE, Polynomial, GCE, BMSE, FL, CBL, Range loss, our constructed loss function, the best results are high-lighted in bold.

	CE	Noisy label robust			Negative sample robust			Ours
		Poly	GCE	BMSE	FL	CBL	Range	
F1	50.98	50.16	49.91	49.56	50.93	51.23	50.87	**51.25**
Ign F1	45.32	44.98	44.25	43.89	46.02	46.05	45.66	**46.13**
AUC	50.06	49.23	49.11	49.23	50.21	50.24	50.26	**50.35**
Ign AUC	46.22	45.11	45.05	44.74	46.18	**46.35**	45.75	46.25

Table 4
F1 score and accuracy score for Dev dataset for model trained in artificially noised dataset of CoNLL 2003 with CE, GCE, BMSE, Polynomial, FL, CBL, Range loss, our constructed loss function, the best results are high-lighted in bold.

Loss functions		Noise rate			
		0.2	0.4	0.6	0.8
CE	F1	85.12	73.23	55.27	44.36
	ACC	82.66	70.55	52.67	41.75
GCE	F1	84.51	73.15	56.04	46.72
	ACC	81.15	70.09	53.60	43.92
BMSE	F1	80.08	71.11	54.96	44.11
	ACC	78.76	68.62	52.06	41.38
Poly	F1	83.17	73.61	55.92	46.93
	ACC	81.01	70.69	53.61	43.15
FL	F1	84.76	73.51	55.36	44.27
	ACC	81.95	70.74	52.47	40.69
CBL	F1	83.30	72.89	55.45	44.55
	ACC	80.25	70.15	52.68	41.47
Range	F1	82.59	73.53	54.65	44.39
	ACC	79.87	70.66	51.51	41.02
Ours	F1	**85.88**	**74.11**	**56.43**	**47.68**
	ACC	**83.16**	**71.63**	**53.38**	**44.03**

negative samples and 34k positive samples (Li, Sun et al., 2020), and is highly imbalanced. In this dataset, $K = 8$, $\bar{\rho} = 2.22e-2$, thus we set $C_1 = 1e{-}3$. Then we add uniform noise to the label of NER task artificially to evaluate the noisy label robustness of our loss function. The noise is added the same as the uniform noised dataset in Docred, and the noise rate are [0.2, 0.4, 0.6, 0.8]. The model we adopted are BERT from Devlin et al. (2019). The evaluation metrics are F1 score and accuracy (ACC) score. The results are shown in Table 4.

As depicted in Table 4, model adopted our loss function achieved higher performance than other models, which proved the generalization of our loss function.

5. Conclusion and discussion

In conclusion, we quantified the negative impact from noisy label and negative sample, and proposed a loss function that can address these two problems to the best extend. In this way, the best model performance under such method can be revealed. The performance of the DNN model adopting our loss function exceeded the DNN models that adopting state-of-the-art noisy label robust and negative sample robust loss functions on distantly labeled dataset, artificially noised dataset, human-annotated dataset of Docred, as well as artificially noised dataset of CoNLL 2003, which proved the effectiveness of our loss function. Although adopting our proposed method requires linear programming, but when set M to a relatively large value (e.g., 50/100), the computational time for our method can be ignored, and set higher M will not influence the model performance.

On the other hand, the loss function proposed in this paper are determined only by the probability assigned to the given label, it is possible to further improve the noise-robustness of the loss function if other labels are considered. Thus, the future work can be focused on the scenario that considered the probability of other label.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Project No. 61977013) and Sichuan Science and Technology Program, China (Project No. 2019YJ0164).

Appendix A

Proof. This is the proof for Lemma 1. According to Jensen Inequality, if $l''(\mathbf{p}_y) \geq 0$ and $l'(\mathbf{p}_y) \leq 0$ on interval $(0, 1)$, then for $\forall \mathbf{x} \in \mathcal{X}$:

$$\sum_{k=0}^{K} \frac{1}{K+1} l(f_k(\mathbf{x}, \theta)) \geq l(\sum_{k=0}^{K} \frac{1}{K+1} f_k(\mathbf{x}, \theta))$$

then:

$$\sum_{k=0}^{K} l(f_k(\mathbf{x}, \theta)) \geq (K+1) l(\frac{\sum_{k=0}^{K} f_k(\mathbf{x}, \theta)}{K+1}) = (K+1) l(\frac{1}{K+1}) \quad (16)$$

Since $l''(\mathbf{p}_y) \geq 0$ in interval $(0, 1)$, then for $\forall p_a, p_b \in (0, 1)$:

$$\int_0^{p_b} l'(\mathbf{p}_y)\, d\mathbf{p}_y \leq \int_{p_a}^{p_a+p_b} l'(\mathbf{p}_y)\, d\mathbf{p}_y$$

Then

$$l(p_b) - l(0) \leq l(p_a + p_b) - (p_a)$$

And

$$l(p_a) + l(p_b) \leq l(p_a + p_b) + l(0)$$

Therefore,

$$\begin{aligned}
&l(f_0(\mathbf{x}, \theta)) + l(f_1(\mathbf{x}, \theta)) + \ldots + l(f_K(\mathbf{x}, \theta)) \\
&\leq l(0) + l(f_1(\mathbf{x}, \theta)) + \ldots + l(f_0(\mathbf{x}, \theta) + f_K(\mathbf{x}, \theta)) \\
&\leq 2l(0) + l(f_2(\mathbf{x}, \theta)) + \ldots + l(f_0(\mathbf{x}, \theta) + f_1(\mathbf{x}, \theta) + f_K(\mathbf{x}, \theta)) \\
&\leq Kl(0) + l(f_0(\mathbf{x}, \theta) + \ldots + f_K(\mathbf{x}, \theta)) \\
&= Kl(0) + l(1)
\end{aligned}$$

Then the lower bound and upper bound for $\sum_{k=0}^{K} l(f_k(\mathbf{x}, \theta))$ are $(K+1) l(\frac{1}{K+1})$ and $Kl(0) + l(1)$. The proof is completed. □

Appendix B

Proof. This is the proof for Theorem 1. Under uniform noise,

$$\begin{aligned}
&R_\ell^\eta(f) \\
&= \mathbb{E}_{\mathcal{D}^\eta}[l(f_{\tilde{y}}(\mathbf{x}, \theta))] \\
&= \mathbb{E}_{\mathcal{D}}[(1-\eta) l(f_y(\mathbf{x}, \theta)) + \frac{\eta}{K} \sum_{k \neq y} l(f_k(\mathbf{x}, \theta))] \\
&= \mathbb{E}_{\mathcal{D}}[(1-\eta) l(f_y(\mathbf{x}, \theta)) - \frac{\eta}{K} l(f_y(\mathbf{x}, \theta)) + \frac{\eta}{K} \sum_{k=0}^{K} l(f_k(\mathbf{x}, \theta))] \\
&= (1 - \eta - \frac{\eta}{K}) R_\ell(f) + \frac{\eta}{K} \mathbb{E}_{\mathcal{D}}[\sum_{k=0}^{K} l(f_k(\mathbf{x}, \theta))] \\
&= \frac{K - \eta(K+1)}{K} R_\ell(f) + \frac{\eta}{K} \mathbb{E}_{\mathcal{D}}[\sum_{k=0}^{K} l(f_k(\mathbf{x}, \theta))]
\end{aligned} \quad (17)$$

Eq. (17) can be transformed as:

$$R_\ell(f) = \frac{K}{K-\eta(K+1)}(R_\ell^\eta(f) - \frac{\eta}{K}\mathbb{E}_{\mathcal{D}}[\sum_{k=0}^{K} l(f_k(\mathbf{x},\theta))]) \tag{18}$$

Then $R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*)$ is:

$$\begin{aligned} R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*) = &\frac{K}{K-\eta(K+1)}(R_\ell^\eta(\hat{f}_\ell^*) - R_\ell^\eta(f_\ell^*)) \\ &- \frac{\eta}{K}\mathbb{E}_{\mathcal{D}}[\sum_{k=0}^{K} l(\hat{f}_k^*\ell(\mathbf{x},\theta)) - l(f_k^*\ell(\mathbf{x},\theta))]) \end{aligned} \tag{19}$$

Since $\hat{f}_\ell^*$ is the minimizer of $R_\ell^\eta(f)$, then we have:

$$R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*) \le \frac{\eta\mathbb{E}_{\mathcal{D}}[\sum_{k=0}^{K} l(\hat{f}_k^*\ell(\mathbf{x},\theta)) - l(f_k^*\ell(\mathbf{x},\theta))]}{K-\eta(K+1)} \tag{20}$$

According to Lemma 1, If $l''(f_y(\mathbf{x},\theta)) \ge 0$ and $l'(f_y(\mathbf{x},\theta)) \le 0$ on interval (0,1), and $l(1) = 0$, then Eq. (20) can be transformed as:

$$R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*) \le \frac{\eta(Kl(0) + l(1) - (K+1)l(\frac{1}{K+1}))}{K-\eta(K+1)} \tag{21}$$

Since f_ℓ^* is the minimizer of $R_\ell(f)$, then $R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*)$ is lower bounded by 0. The proof is completed. □

Appendix C

Proof. This is the proof for Theorem 2. First we have:

$$R_\ell^\eta(f) = \mathbb{E}_{\mathcal{D}}[(1-\sum_{j\neq y}\eta_{yj})(l(f_y(\mathbf{x},\theta)))] + \mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}(l(f_j(\mathbf{x},\theta)))] \tag{22}$$

Then

$$\begin{aligned} R_\ell^\eta(\hat{f}^*) - R_\ell^\eta(f^*) = &\mathbb{E}_{\mathcal{D}}[(1-\sum_{j\neq y}\eta_{yj})(l(\hat{f}_y^*(\mathbf{x},\theta)) - l(f_y^*(\mathbf{x},\theta)))] \\ &+ \mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}(l(\hat{f}_j^*(\mathbf{x},\theta)) - l(f_j^*(\mathbf{x},\theta)))] \end{aligned} \tag{23}$$

Since $\hat{f}^*$ is the minimizer of $R_\ell^\eta(f)$, then $R_\ell^\eta(\hat{f}^*) - R_\ell^\eta(f^*) \le 0$. Therefore,

$$\begin{aligned} &\mathbb{E}_{\mathcal{D}}[(1-\sum_{j\neq y}\eta_{yj})(l(\hat{f}_y^*(\mathbf{x},\theta)) - l(f_y^*(\mathbf{x},\theta)))] \\ &\le -\mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}(l(\hat{f}_j^*(\mathbf{x},\theta)) - l(f_j^*(\mathbf{x},\theta)))] \end{aligned} \tag{24}$$

When $R_\ell(f^*) = 0$, $l(f_y^*(\mathbf{x},\theta)) = l(1) = 0$ and when $j \neq y$, $l(f_j^*(\mathbf{x},\theta)) = l(0)$. Then Eq. (24) can be transformed as:

$$\begin{aligned} &\mathbb{E}_{\mathcal{D}}[(1-\sum_{j\neq y}\eta_{yj})l(\hat{f}_y^*(\mathbf{x},\theta))] \le l(0)\mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}] \\ &- \mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}l(\hat{f}_j^*(\mathbf{x},\theta))] \end{aligned} \tag{25}$$

Since $\mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}l(\hat{f}_j^*(\mathbf{x},\theta))] \ge 0$, then we have:

$$\mathbb{E}_{\mathcal{D}}[(1-\sum_{j\neq y}\eta_{yj})l(\hat{f}_y^*(\mathbf{x},\theta))] \le l(0)\mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}] \tag{26}$$

Since $min_{y=0,1,\dots,K}(1-\sum_{j\neq y}\eta_{yj})\mathbb{E}_{\mathcal{D}}[l(\hat{f}_y^*(\mathbf{x},\theta))] \le \mathbb{E}_{\mathcal{D}}[(1-\sum_{j\neq y}\eta_{yj})l(\hat{f}_y^*(\mathbf{x},\theta))]$, then Eq. (26) can be transformed as:

$$\min_{y=0,1,\dots,K}(1-\sum_{j\neq y}\eta_{yj})\mathbb{E}_{\mathcal{D}}[l(\hat{f}_y^*(\mathbf{x},\theta))] \le l(0)\mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}] \tag{27}$$

Then

$$\mathbb{E}_{\mathcal{D}}[l(\hat{f}_y^*(\mathbf{x},\theta))] = R_\ell(\hat{f}_\ell^*) \le \frac{l(0)\mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}]}{min_{y=0,1,\dots,K}(1-\sum_{j\neq y}\eta_{yj})} \tag{28}$$

Since $R_\ell(f_\ell^*) \ge 0$ Therefore,

$$R_\ell(\hat{f}_\ell^*) - R_\ell(f_\ell^*) \le \frac{l(0)\mathbb{E}_{\mathcal{D}}[\sum_{j\neq y}\eta_{yj}]}{min_{y=0,1,\dots,K}(1-\sum_{j\neq y}\eta_{yj})} \tag{29}$$

The proof is completed. □

Appendix D

Proof. This is the proof for Proposition 1. Set θ^t be the model's parameters in the tth batch. Since the model is randomly initialized, expectation for predicted probability for every label is $\frac{1}{K+1}$. And $\frac{\partial \mathbf{u}_j(\mathbf{x},\theta^t)}{\partial\theta}$ with different input $\mathbf{x}$ can be approximated as equal when the DNN model did not possess discriminant ability for label j, according to Eq. (11), the expectation of gradient update associated with label j in the first batch is:

$$\begin{aligned} &\mathbb{E}_{\mathcal{D}}[\mathbf{G}_j(\mathcal{D},\theta^1)] \\ &= N\Big(\mathbb{P}[y=j]\mathbb{E}_{\mathbf{x}|y=j}[l'(\frac{1}{K+1})\frac{1}{K+1}(1-\frac{1}{K+1})] \\ &\quad - \sum_{k\neq j}\mathbb{P}[y=k]\mathbb{E}_{\mathbf{x}|y=k}[l'(\frac{1}{K+1})\frac{1}{(K+1)^2}]\Big)\,\mathbb{E}_{\mathcal{D}}[\frac{\partial \mathbf{u}_j(\mathbf{x},\theta^1)}{\partial\theta}] \\ &= Nl'(\frac{1}{K+1})\Big(\mathbb{P}[y=j]\frac{1}{K+1} - \sum_{k=0}^{K}\mathbb{P}[y=k]\frac{1}{(K+1)^2}\Big) \\ &= Nl'(\frac{1}{K+1})\frac{\mathbb{P}[y=j] - \sum_{k=0}^{K}\frac{\mathbb{P}[y=k]}{K+1}}{K+1}\mathbb{E}_{\mathcal{D}}[\frac{\partial \mathbf{u}_j(\mathbf{x},\theta^1)}{\partial\theta}] \\ &= Nl'(\frac{1}{K+1})\frac{\mathbb{P}[y=j] - \frac{1}{K+1}}{K+1}\mathbb{E}_{\mathcal{D}}[\frac{\partial \mathbf{u}_j(\mathbf{x},\theta^1)}{\partial\theta}] \end{aligned} \tag{30}$$

Since $l'(\frac{1}{K+1}) < 0$, when $\mathbb{P}[y=j] < \frac{1}{K+1}$, then $\mathbf{g}_j(\mathcal{D},\theta^1) = Nl'(\frac{1}{K+1})\frac{\mathbb{P}[y=j]-\frac{1}{K+1}}{K+1} > 0$, the gradient update will tend to decrease the prediction accuracy for label j. The proof is completed. □

Appendix E

Proof. The proof for Proposition 2 is as follows: under Assumption 1, when ignore the difference of $\frac{\partial \mathbf{u}_j(\mathbf{x},\theta)}{\partial\theta}$ between different input $\mathbf{x}$ in the tth batch, Eq. (11) can be transformed as:

$$\begin{aligned} &\mathbb{E}_{\mathcal{D}}[\mathbf{G}_j(\mathcal{D},\theta^t)] \\ &= N\Big(\mathbb{P}[y=j]\mathbb{E}_{\mathbf{x}|y=j}[l'(f_j(\mathbf{x},\theta^t))(1-f_j(\mathbf{x},\theta^t))f_j(\mathbf{x},\theta^t)] - \\ &\sum_{k\neq j}\mathbb{P}[y=k]\mathbb{E}_{\mathbf{x}|y=k}[l'(f_k(\mathbf{x},\theta^t))f_k(\mathbf{x},\theta^t)f_j(\mathbf{x},\theta^t)]\Big)\mathbb{E}_{\mathcal{D}}[\frac{\partial \mathbf{u}_j(\mathbf{x},\theta^t)}{\partial\theta}] \\ &= N\Big(\mathbb{P}[y=j]\mathbb{E}_{\mathbf{x}|y=j}[l'(f_j(\mathbf{x},\theta^t))f_j(\mathbf{x},\theta^t)] \\ &- \sum_{k=0}^{K}\mathbb{P}[y=k]\mathbb{E}_{\mathbf{x}|y=k}[l'(f_k(\mathbf{x},\theta^t))f_k(\mathbf{x},\theta^t)f_j(\mathbf{x},\theta^t)]\Big)\mathbb{E}_{\mathcal{D}}[\frac{\partial \mathbf{u}_j(\mathbf{x},\theta^t)}{\partial\theta}] \end{aligned} \tag{31}$$

When ignoring the difference of $f_j(\mathbf{x},\theta^t)$ between different input $\mathbf{x}$, Eq. (31) can be transformed as:

$$\begin{aligned} &\mathbb{E}_{\mathcal{D}}[\mathbf{G}_j(\mathcal{D},\theta^t)] \\ &= N\mathbb{E}_{\mathcal{D}}[f_j(\mathbf{x},\theta^t)]\Big(\mathbb{P}[y=j]l'(\mathbb{E}_{\mathbf{x}|y=j}[f_j(\mathbf{x},\theta^t)]) \end{aligned}$$

$$-\sum_{k=0}^{K}\mathbb{P}[y=k]\mathbb{E}_{\mathbf{x}|y=k}[l'(f_k(\mathbf{x},\theta^t))f_k(\mathbf{x},\theta^t)]\Big)\mathbb{E}_{\mathcal{D}}[\frac{\partial \mathbf{u}_j(\mathbf{x},\theta^t)}{\partial\theta}] \tag{32}$$

Since $(l'(p)p)' \leq 0$ when $p \in (0,1)$, then we have:

$$\begin{aligned}
&\mathbf{g}_j(\mathcal{D},\theta^t)\\
&= N\mathbb{E}_{\mathcal{D}}[f_j(\mathbf{x},\theta^t)]\Big(\mathbb{P}[y=j]l'(\mathbb{E}_{\mathbf{x}|y=j}[f_j(\mathbf{x},\theta^t)])\\
&\quad -\sum_{k=0}^{K}\mathbb{P}[y=k]\mathbb{E}_{\mathbf{x}|y=k}[l'(f_k(\mathbf{x},\theta^t))f_k(\mathbf{x},\theta^t)]\Big)\\
&\leq N\mathbb{E}_{\mathcal{D}}[f_j(\mathbf{x},\theta^t)]\ \Big(\ \mathbb{P}[y=j]l'(\mathbb{E}_{\mathbf{x}|y=j}[f_j(\mathbf{x},\theta^t)])\\
&\quad - l'(1)\sum_{k=0}^{K}\mathbb{P}[y=k]\ \Big)\\
&= N\mathbb{E}_{\mathcal{D}}[f_j(\mathbf{x},\theta^t)]\Big(\mathbb{P}[y=j]l'(\mathbb{E}_{\mathbf{x}|y=j}[f_j(\mathbf{x},\theta^t)]) - l'(1)\Big)
\end{aligned} \tag{33}$$

Let $p_j = \mathbb{E}_{\mathbf{x}|y=j}[f_j(\mathbf{x},\theta^t)]$, and $U_j(p_j) = \mathbb{P}[y=j]l'(p_j) - l'(1)$. Since $l'(0) \to -\infty$, then there exists a solution $\mu_j \in (0, \frac{1}{K+1})$ for $U_j(p)=0$, and when $p_j < \mu_j$, $\mathbf{g}_j(\mathcal{D},\theta^t) < 0$, the prediction accuracy for label j will be increased. Therefore μ_j is the turning value for label j. This complete the proof. □

Appendix F

Proof. This is the proof for Lemma 2. When $p \in (0,1)$, $A_n \leq 0, n = 1,\ldots,M$, and $D(1) = \sum_{n=1}^{d}A_n - \sum_{m=d+1}^{M}A_m \leq 0$, since $F(n)$ is a monotone increasing function for n, we have:

$$\begin{aligned}
&D(p)\\
&= \sum_{n=1}^{d}A_n p^{F(n)} - \sum_{m=d+1}^{M}A_m p^{F(m)}\\
&\leq p^{F(d)}\sum_{n=1}^{d}A_n - p^{F(d)}\sum_{m=d+1}^{M}A_m\\
&\leq \sum_{n=1}^{d}A_n - \sum_{m=d+1}^{M}A_m\\
&= D(1)\\
&\leq 0
\end{aligned} \tag{34}$$

This complete the proof. □

References

Amid, E., Warmuth, M. K., Anil, R., & Koren, T. (2019). Robust bi-tempered logistic loss based on bregman divergences. In *33rd conference on neural information processing systems (NIPS), Vol. 32*.

Bartlett, P. L., Jordan, M. I., & Mcauliffe, J. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, *101*(473), 138–156.

Chen, Z., Shen, S., Hu, Z., Lu, X., Mei, Q., & Liu, X. (2019). Emoji-powered representation learning for cross-lingual sentiment classification. In *Proceedings of the 28th international conference on world wide web (WWW)* (pp. 251–262).

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 9260–9269).

Dasigi, P., Liu, N., Marasovic, A., Smith, N., & Gardner, M. (2019). Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 conference on empirical methods in natural language processing (EMNLP)* (pp. 5925–5932).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics* (pp. 4171–4186).

Dong, X., He, S., & Stojanovic, V. (2020). Robust fault detection filter design for a class of discrete-time conic-type nonlinear Markov jump systems with jump fault signals. *IET Control Theory & Applications*, 1912–1919.

Fu, T. J., & Ma, W. Y. (2019). Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th annual meeting of the association for computational linguistics (ACL)* (pp. 1409–1418).

Geng, Z., Chen, G., Han, Y., Lu, G., & Li, F. (2020). Semantic relation extraction using sequential and tree-structured lstm with attention. *Information Sciences*, *509*, 183–192.

Ghosh, A., Kumar, H., & Sastry, P. S. (2017). Robust loss functions under label noise for deep neural networks. In *Proceedings of the thirty-first AAAI conference on artificial intelligence* (pp. 1919–1925).

Gong, M., Li, H., Meng, D., Miao, Q., & Liu, J. (2019). Decomposition-based evolutionary multiobjective optimization to self-paced learning. *IEEE Transactions on Evolutionary Computation*, *23*, 288–302.

Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., & Sun, M. (2018). FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP)* (pp. 4803–4809).

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, *4*, 373–395.

Li, Y., Long, G., Shen, T., Zhou, T., & Jiang, J. (2020). Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 8269–8276).

Li, X., Sun, X., Meng, Y., Liang, J., Wu, F., & Li, J. (2020). Dice loss for data-imbalanced nlp tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics (ACL)* (pp. 465–476).

Lin, T., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 2999–3007).

Lin, Y., Shen, S., Liu, Z., Luan, H., & Sun, M. (2016). Neural relation extraction with selective attention over instances. In *Proceedings of the 54th annual meeting of the association for computational linguistics (ACL)* (pp. 2124–2133).

Liu, J., Shou, L., Pei, J., Gong, M., Yang, M., & Jiang, D. (2020). Cross-lingual machine reading comprehension with language branch knowledge distillation. In *Proceedings of the 28th international conference on computational linguistics (COLING)* (pp. 2710–2721).

Luo, B., Feng, Y., Wang, Z., Zhu, Z., Huang, S., Yan, R., & Zhao, D. (2017). Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix. In *Proceedings of the 55th annual meeting of the association for computational linguistics (ACL)* (pp. 430–439).

van der Maaten, L., & Hinton, G. (2008). Visualizing data using T-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Marcelino, G., Pinto, R., & Magalhães, J. a. (2018). Ranking news-quality multimedia. In *Proceedings of the 2018 ACM on international conference on multimedia retrieval (ICMR)* (pp. 10–18).

Namysl, M., Behnke, S., & Köhler, J. (2020). Nat: Noise-aware training for robust neural sequence labeling. In *Proceedings of the 58th annual meeting of the association for computational linguistics (ACL)* (pp. 1501–1517).

Qu, J., Ouyang, D., Hua, W., Ye, Y., & Li, X. (2018). Distant supervision for neural relation extraction integrated with word attention and property features. *Neural Networks*, *100*, 59–69.

Ren, X., Wu, Z., He, W., Qu, M., Voss, C. R., Ji, H., Abdelzaher, T., & Han, J. (2017). CoType: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th international conference on world wide web (WWW)* (pp. 1015–1024).

Riedel, S., Yao, L., & Mccallum, A. (2010). Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European conference on machine learning and knowledge discovery in databases (ECML)* (pp. 148–163).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Stojanovic, V., He, S., & Zhang, B. (2020). State and parameter joint estimation of linear stochastic systems in presence of faults and non-Gaussian noises. *International Journal of Robust and Nonlinear Control*, (4), 6683–6700.

Stojanovic, V., & Prsic, D. (2020). Robust identification for fault detection in the presence of non-Gaussian noises: application to hydraulic servo drives. *Nonlinear Dynamics*, *100*(4), 2299–2313.

Sun, C., Gong, Y., Duan, N., Gong, M., Jiang, D., Sun, S., Lan, M., & Wu, Y. (2019). Joint type inference on entities and relations via graph convolutional networks. In *Proceedings of the 57th annual meeting of the association for computational linguistics (ACL)* (pp. 1361–1370).

Takanobu, R., Zhang, T., Liu, J., & Huang, M. (2019). A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 7072–7079).

Tao, H., Wang, P., Chen, Y., Stojanovic, V., & Yang, H. (2020). An unsupervised fault diagnosis method for rolling bearing using stft and generative neural networks. *Journal of the Franklin Institute*, *357*(11), 7286–7307.

Vaidya, P. M. (1989). Speeding-up linear programming using fast matrix multiplication. In *30th annual symposium on foundations of computer science* (pp. 332–337).

Vashishth, S., Joshi, R., Prayaga, S. S., Bhattacharyya, C., & Talukdar, P. (2018). Reside: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP)* (pp. 1257–1266).

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., & Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)* (pp. 322–330).

Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. *Pattern Recognition Letters*, *9911*, 499–515.

Wu, S., Fan, K., & Zhang, Q. (2019). Improving distantly supervised relation extraction with neural noise converter and conditional optimal selector. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 7273–7280).

X. Zhang, Z. Fang, Wen, Y., Li, Z., & Qiao, Y. (2017). Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE international conference on computer vision (ICCV)* (pp. 5409–5418).

Xu, P., & Barbosa, D. (2019). Connecting language and knowledge with heterogeneous representations for neural relation extraction. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies (NAACL:HLT)* (pp. 3201–3206).

Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., & Sun, M. (2019). DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th annual meeting of the association for computational linguistics (ACL)* (pp. 764–777).

Ye, H., & Luo, Z. (2019). Deep ranking based cost-sensitive multi-label learning for distant supervision relation extraction. *Information Processing and Management*, Article 102096.

Zeng, W., Lin, Y., Liu, Z., & Sun, M. (2017). Incorporating relation paths in neural relation extraction. In *Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP)* (pp. 1768–1777).

Zeng, D., Liu, K., Chen, Y., & Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP)* (pp. 1753–1762).

Zhang, W., Chen, Y., Yang, W., Wang, G., Xue, J., & Liao, Q. (2020). Class-variant margin normalized softmax loss for deep face recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 1–6.

Zhang, Y., Guo, Z., & Lu, W. (2019). Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th annual meeting of the association for computational linguistics (ACL)* (pp. 241–251).

Zhang, Z., & Sabuncu, M. R. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd conference on neural information processing systems (NIPS)* (pp. 8792–8802).

Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., & Xu, B. (2017). Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th annual meeting of the association for computational linguistics (ACL)* (pp. 1227–1236).

Zhou, L., Tao, H., Paszke, W., Stojanovic, V., & Yang, H. (2020). Pd-type iterative learning control for uncertain spatially interconnected systems. *Mathematics*, *1528*(9).

Zhou, P., Xu, J., Qi, Z., Bao, H., Chen, Z., & Xu, B. (2018). Distant supervision for relation extraction with hierarchical selective attention. *Neural Networks*, *108*, 240–247.