Partial Label Learning Based on Fully Connected Deep Neural Network

Houjie Li¹, Lei Wu¹, Jianjun He¹, Ruirui Zheng¹, Yu Zhou¹, Shuang Qiao²
¹College of Information and Communication Engineering, Dalian Minzu University, Dalian, China
²School of Management, Dalian Polytechnic University, Dalian, China
*Corresponding author, Email address: qiaoshuang@sina.com

Received: June 10, 2021. Revised: December 14, 2021. Accepted: January 10, 2022. Published: January 12, 2022.

Abstract- The ambiguity of training samples in the partial label learning framework makes it difficult for us to develop learning algorithms and most of the existing algorithms are proposed based on the traditional shallow machine learning models, such as decision tree, support vector machine, and Gaussian process model. Deep neural networks have demonstrated excellent performance in many application fields, but currently it is rarely used for partial label learning framework. This study proposes a new partial label learning algorithm based on a fully connected deep neural network, in which the relationship between the candidate labels and the ground-truth label of each training sample is established by defining three new loss functions, and a regularization term is added to prevent overfitting. The experimental results on the controlled UCI datasets and real-world partial label datasets reveal that the proposed algorithm can achieve higher classification accuracy than the state-of-the-art partial label learning algorithms.

Keywords- Partial label learning, Deep neural network, Weakly annotated data.

I. INTRODUCTION

Traditional supervised learning frameworks have demonstrated their excellent performance in many applications, however, their algorithms need to be trained on the samples with strong supervision information, which implies that the label of each training sample must be annotated accurately. However, with the rapid development of information technology and the arrival of the era of big data, large amounts of data are not labeled or are inaccurately labeled. Furthermore, manual annotation is time-consuming and labor-intensive. Therefore, it is necessary to develop machine learning algorithms for weakly annotated data.

Partial label learning is an important weakly supervised learning framework [1] dealing with the problem where each training sample is associated with a set of candidate labels, only one of which is the ground-truth label. In comparison with the precisely annotated data required by traditional supervised learning frameworks, it is obvious that partial label data are easier to obtain. Therefore, partial label learning has become a hot research topic of machine learning community and has been widely used in various practical problems such as image annotation [2], web mining [3] and age estimation [4]. However, since the ground-truth label of the training sample is concealed in a candidate label set, it is difficult to develop learning model. At present, only a few partial label learning algorithms have been proposed and most of them are based on traditional shallow machine learning models such as K-nearest neighbor method [5], support vector machines [1, 7], maximum margin method [2, 5], logistic regression model [9], decision trees [10, 11], Graph model [12].

Owing to the excellent performance in the fields of computer vision [13], speech recognition [14], natural language processing [15], and image retrieval [16, 17], deep neural networks have become a widely used tool for developing learning algorithms. Moreover, in recent years, there have been various attempts to provide a theoretical foundation for the representation of power, optimization, and generalization of deep neural networks. Reference [18] demonstrated that a deep neural network could divide the space into an exponential number of sets, which cannot be achieved by shallow networks using the same number of parameters. Cohen et al. [19] proved that functions implemented by deep neural networks are exponentially more expressive than functions implemented by shallow networks, and then Bentkamp et al. [20] formalized their mathematical proof using Isabelle/HOL. Sun et al. [21] showed that deeper neural networks tend to have larger representation power (measured by Betti numbers based complexity) than shallower networks in a multi-class setting, and thus could lead to smaller empirical margin errors. Thus, both theory and practice indicate that deep neural networks are superior to the traditional shallow machine learning models. However, till now, it is rarely used to solve partial label learning problems. In view of this, this study proposes a new partial label learning algorithm named PL-DNN based on a fully
connected deep neural network, in which the relationship between the candidate labels and the ground-truth label of each training sample is established by defining three new loss functions, and a regularization term is added to prevent overfitting. The main contributions are as follows:

(1) A new deep partial label learning algorithm is proposed by using a fully connected deep neural network.

(2) Because three different loss functions are defined, the proposed algorithm can better describe the relationship between the candidate labels and the ground-truth label of the training samples.

II. RELATED WORK

The partial label learning framework is also termed as superset label learning [22,23] or ambiguous label learning[24,25]. The earliest research on this framework can date back to the work of Grandvalet [9]. Propelled by the works of Grandvalet, it began to attract more and more attention. Hillermeier et al. [5] and Zhang et al. [6] proposed two partial label learning algorithms based on the K-nearest neighbor method. Cour et al. [10] and Nguyen et al. [7] proposed a partial label learning algorithm based on support vector machine by extending the traditional two-classification and multi-classification loss functions, respectively. Luo et al. [2] and Yu et al. [1] proposed two partial margin partial label learning algorithms. Grandvalet et al. [1] proposed a partial label learning algorithm based on the logistic regression model. Beygelzimer et al. [10] and Yu et al. [11] proposed two partial label learning algorithms based on decision trees. Lyu et al. [26] and Wang et al. [27] proposed two partial label learning algorithms based on Graph model. Feng et al. [28] developed a partial label learning algorithm by using a self-training strategy. Xu et al. [29] proposed a partial label learning algorithm based on label enhancement strategy. Lyu et al. [28] and Wang et al. [29] developed two self-paced strategies to address the partial label learning problems. Ly et al. [30] and Yao et al. [31] proposed two partial label learning algorithms by progressively identifying the ground-true label. Feng and An [32] proposed a label distribution based approach to address the partial label learning problems. Gong et al. [33] proposed a novel regularization approach for instance-based partial label learning. Wu et al. [34] proposed a partial label learning algorithm based on a binary decomposition approach. Feng et al. [35] proposed a partial label learning approach by maximizing the latent semantic differences between two samples. Yan and Guo [36] proposed a partial label learning algorithm with batch label correction. Wang et al. [37] proposed a learning algorithm for the class-imbalance partial label learning problem. Xie et al. [38] and Yan et al. [39] studied the partial multi label learning problem. Zhou et al. [40] and Xu et al. [41] investigated the metric learning problem of partial label data. It can be seen that almost all of the existing partial label learning algorithms are developed by using the traditional shallow machine learning models such as K-nearest neighbor model, support vector machine and decision tree.

Because the label information of training samples in the partial label learning framework is ambiguous and not unique, to extend the traditional classification model to the partial label learning framework, the main problem should be solved is to define a loss function that can describe the relationship between the candidate labels and the ground-truth label of the training samples. The existing work mainly focused on two types of loss functions: the average loss function and maximum loss function. The average loss function assigns the same weight to each candidate label, which essentially assumes that, for a certain training sample, all of the candidate labels are its ground-truth label. Thus, the average loss functions do not accord with the definition of partial label learning framework who states that one and only one candidate label is the ground-truth label of each training sample. Despite the inaccurate assumption, average loss functions are widely used by many algorithms to solve the partial label learning problem because of their simple form. For example, Cour et al. [42] defined an average loss function to develop a SVM (support vector machine) based partial label learning algorithm. In [43], the logistic regression model was used to establish the partial label learning algorithm with an average loss function. The maximum loss function can better describe the relationship between the candidate labels and the ground-truth label of the training sample. However, the model with maximum loss function is usually difficult to solve, and only a few partial label learning algorithms have been proposed using such loss functions thus far. Nguyen et al. [44] proposed a partial label learning algorithm based on SVM by using a maximum loss function; Zhou et al. [45] defined a maximum loss function to develop the Gaussian process model based partial label learning algorithm. Because these two types of loss functions have their own advantages and disadvantages, our proposed PL-DNN algorithm will adopt both of them.

III. THE PL-DNN ALGORITHM

A. Network structure and loss function

Let $X = \mathbb{R}^d$ be the $d$-dimensional instance space and $Y = \{1, 2, \ldots, q\}$ be the label space with $q$ class labels. Given a partial label training set $D = \{(x_i, S_i)\mid 1 \leq i \leq n\}$, where $x_i \in X$ is the feature vector of the $i$th training sample and $S_i \subseteq Y$ is the corresponding candidate label set. The ground-truth label of $x_i$ is concealed in the candidate label set $S_i$ and is inaccessible to the learning algorithm. The purpose of partial label learning framework is to learn a multi-class classifier $f : X \to Y$ based on training set $D$. This study proposes a partial label learning algorithm based on deep neural network model. Figure 1 illustrates the basic structure of the fully connected deep neural network used in this study. For a given input $x \in X$, the output of the first hidden layer is $h_1 = \varphi(W_1^T x + b_1)$, where $W_1$ is the weight between the input layer and the first hidden layer, $b_1$ is the bias parameter of the first hidden layer and $\varphi(\cdot)$ is the activation function. The activation func-
The output of the \( m \)th hidden layer is \( h_m = \varphi(W^T_m h_{m-1} + b_m) \), where \( h_{m-1} \) is the output of the \((m-1)\)th hidden layer. The value of the output layer is \( y = W^T_m h_m + b_{m+1} \). To facilitate the definition of the loss function, the output layer of the proposed deep neural network does not use any activation function.

Due to the ambiguity of training samples, the traditional deep neural network models cannot directly process partial label learning problems. Therefore, we should define a new loss function to capture the ground-truth label of the training sample from its candidate label set. Two average loss functions and one maximum loss function are defined in this paper. The two average loss functions are shown in equation (1) and (2).

\[
\text{loss}_1 = -\frac{1}{n} \sum_{i=1}^{n} \ln \left( \frac{\sum_{y \in S_i} e^{f_y(x_i)}}{\sum_{j=1}^{q} e^{f_j(x_i)}} \right) \quad (1)
\]

\[
\text{loss}_2 = -\frac{1}{n} \sum_{i=1}^{n} \ln \left( \frac{1}{k_i} \sum_{y \in S_i} e^{f_y(x_i)}}{\sum_{j=1}^{q} e^{f_j(x_i)}} \right) \quad (2)
\]

where \( k_i \) denotes the number of candidate labels in the candidate label set \( S_i \) and \( f_j(x_i) \) is the output of the \( j \)th node in the output layer on sample \( x_i \).

Loss functions (1) and (2) essentially assume that all the candidate labels are the ground-truth label of the training sample, which is inconsistent with the description of the relationship between the ground-truth label and the candidate label set in the partial label learning framework. Thus, a maximum loss function is also defined in equation (3).

\[
\text{loss}_3 = -\frac{1}{n} \sum_{i=1}^{n} \ln \max_{y \in S_i} \left\{ e^{f_y(x_i)} / \sum_{j=1}^{q} e^{f_j(x_i)} \right\} \quad (3)
\]

The maximum loss function assumes that the candidate label with maximum probability is the ground-truth label of each training sample, which can better capture the relationship between the ground-truth label and the candidate label set. As the max \{ \cdot \} function is a non-smooth function, the model with loss function (3) cannot be solved by gradient-based methods. Therefore, inspired by [43], the loss function (3) is approximated by a smooth function using the aggregate function defined in equation (4). The aggregate function

\[
G_l(x) = \frac{1}{l} \ln \left( \sum_{j=1}^{k} e^{g_j(x)} \right) \quad (4)
\]

was derived by Li[44] from the Jaynes maximum entropy principle when solving nonlinear programming problems, where, \( l \) is a positive control parameter.

The aggregate function \( G_l(x) \) and the maximum function \( G(x) = \max_{j \in \{1,2,\ldots,k\}} g_j(x) \) exhibit the following relationship

\[
G(x) \leq G_l(x) \leq G(x) + \frac{\ln k}{l} \quad (5)
\]
Thus, \( G(x) \) can be approximated uniformly by \( G_l(x) \) when \( l \to +\infty \). According to (5), the following two formulas may be obtained

\[
\max_{y \in S_i} f_y(x_i) = \frac{1}{l} \ln \left( \sum_{y \in S_i} e^{f_y(x_i)} \right) 
\leq \max_{y \in S_i} f_y(x_i) + \frac{\ln q}{l}
\]

(6)

\[
\sum_{i=1}^{n} \max_{y \in S_i} f_y(x_i) = \frac{1}{l} \sum_{i=1}^{n} \ln \left( \sum_{y \in S_i} e^{f_y(x_i)} \right) 
\leq \sum_{i=1}^{n} \max_{y \in S_i} f_y(x_i) + \frac{n \ln q}{l}
\]

(7)

It can be seen from (6) and (7) that when \( l \to +\infty, \sum_{i=1}^{n} \max_{y \in S_i} f_y(x_i) \) can be approximated uniformly by

\[
\frac{1}{l} \sum_{i=1}^{n} \ln \left( \sum_{y \in S_i} e^{f_y(x_i)} \right)
\]

Thus, equation (3) can be approximated as follows

\[
\text{loss}_3 = -\frac{1}{n} \sum_{i=1}^{n} \ln \left( \sum_{y \in S_i} e^{f_y(x_i)} \right) 
\approx -\frac{1}{n} \sum_{i=1}^{n} \ln \left( \sum_{y \in S_i} e^{f_y(x_i)} \right) + \frac{n \ln q}{l}
\]

(8)

which is a smooth function. Thus, the model with loss function (8) can be solved by any gradient-based method.

B. Regularization and parameter tuning strategies

To avoid overfitting, two regularization terms, referred to as L2 regularization [35] and dropout regularization [40], are added to the model. L2 regularization means that the squared sum of weights is added to the loss function, thus, the original loss functions (1), (2), and (8) are redefined as follows

\[
\text{loss}_1 = \text{loss}_1 + \lambda \sum_{j=1}^{m+1} W_j^T W_j
\]

(9)

\[
\text{loss}_2 = \text{loss}_2 + \lambda \sum_{j=1}^{m+1} W_j^T W_j
\]

(10)

\[
\text{loss}_3 = \text{loss}_3 + \lambda \sum_{j=1}^{m+1} W_j^T W_j
\]

(11)

where \( \lambda \) is a regularization coefficient. L2 regularization makes the weights of the deep neural network as small as possible, therefore, the overfitting can be avoided to some extent.

Dropout regularization can also be used to avoid overfitting when training deep neural networks. In each training batch, by randomly dropping the neuron nodes of the deep neural network (i.e. to set the output of the node to 0), overfitting can be significantly reduced because it decays the interaction between feature detectors (neuron nodes) and makes the deep neural network simpler and more accurate. In this study, 50% of the neuron nodes in each layer are dropped.

In a deep neural network, the numbers of hidden layers and neuron nodes have significant influence on classification accuracy. It is time consuming to adjust the parameters in a greedy way. Therefore, the following strategies are adopted to select the numbers of hidden layers and neuron nodes. For a given dataset, the input layer and output layer of the neural network are first determined. The number of input layer neurons is the feature dimension of the sample, and the number of neurons in the output layer is the number of classes. Next, the number of hidden layers is selected from 0 to 10, starting from layer 0, and increasing layer by layer. When the number of hidden layers is small, we determine the number of hidden layer neurons based on the data dimension of the dataset. When the feature dimension of the sample is larger than the number of classes, the number of hidden layer neurons decreases, layer by layer, whereas when the feature dimension of the sample is smaller than the number of classes, the number of hidden layer neurons increases, layer by layer. When the number of hidden layers is large, the number of neurons in the hidden layer increases layer by layer and then decreases layer by layer.

IV. Experiments

A. Experimental setup

This study aims to propose a deep neural network based partial label learning algorithm (PL-DNN). To verify the classification accuracy of the algorithm, we conduct experiments on six UCI datasets and five real-world partial label datasets. The specific information of these datasets is shown in Table 1, and the PL-DNN algorithm proposed in this study is compared with six well-known existing partial label learning algorithms. These six partial label learning algorithms include two algorithms proposed by using the k-nearest neighbor model (PL-KNN [5] and PL-LEAF algorithm [4]), a kernel based algorithm PL-ECOC [47], a linear algorithm CLPL [1], an instance-based algorithm RegISL [34], and a binary decomposition algorithm PALOC [35]. The parameters of the first four algorithms are set similar to those in reference [41]. For the PL-KNN algorithm, its parameter \( k \) is selected from the set 5, 8, 11, 14, 17 by cross-validation method. For the PL-LEAF algorithm, its parameters are set with the default values given in [4] as they have little influence on the performance of the algorithm. For the PL-ECOC algorithm, the RBF kernel function \( k(x, x') = \exp(-\gamma ||x-x'||^2) \) is used as the kernel function of the two-class classifier, and the parameters \( \gamma \) and \( k \) are selected from sets \( \{3^{-3}/\delta, 3^{-2}/\delta, \cdots 3^{1}/\delta\} \), and \( \{10^{-1}, 10^{0}, \cdots 10^{1}\} \) by the cross-validation method, respectively, where \( \delta \) is the average distance of the train-
For our PL-DNN algorithm, the control parameter values are set similarly to those in reference [34] and [35]. For the latter two algorithms, the parameter settings are consistent with those in [1].

The model with hinge loss is used as its basic learning model, similar to [47]. For the CLPL algorithm, the SVM is used for classification, and the values of the other parameters are also consistent with the previous work.

In the classification process, for each given pair \((p, r)\), \(p\) denotes the proportion of the partial label sample in the entire sample set, while \(r\) is the number of candidate labels of each partial label sample except its ground-truth label. The transformation process is as follows: for each given pair \((p, r)\), \(pn\) samples are selected from the original UCI dataset first \((n\) is the number of samples in the dataset). Then, for each selected sample, \(r\) labels are randomly selected from the label space except its ground-truth label, and the candidate label set of this sample is composed of the \(r\) selected labels and the ground-truth label. In this section, the experiments are conducted using two values of \(p\) and \(r\), respectively, \(p = 0.3\) or \(0.6\), and \(r = 1\) or \(3\). Therefore, for each UCI dataset, we can obtain four partial label datasets generated according to different \((p, r)\) combinations. Twenty-four partial label datasets can be generated in total, hereafter referred to as the controlled UCI datasets.

Table 2 presents the experimental results of each partial label learning algorithm on the 24 controlled UCI datasets. From the experimental results, we can clearly see that on 22 out of the 24 controlled UCI datasets, the classification accuracy of the proposed PL-DNN algorithm is higher than that of the six existing partial label learning algorithms. On the four controlled datasets generated based on the Movement dataset, the classification accuracy is improved by at least 8.41%. Moreover, it can be seen from the results in Table 2 that the PL-DNN algorithm with \(\text{loss}_2\) loss function exhibits a better classification accuracy on the controlled UCI datasets.

To gain a clear understanding of the relative performance of the PL-DNN algorithm against other algorithms, Table 3 presents the win/tie/loss counts between PL-DNN+Loss2 and the other six algorithms on each controlled UCI dataset based on the right-tailed t-test. It can be seen from Table 3 that the PL-DNN algorithm outperforms other algorithms on most controlled UCI datasets. As a whole, in 100%, 88%, 83%, 100%, 92%
| Data sets | Algorithms    | Accuracy (mean $\pm$ std.) |
|-----------|---------------|-----------------------------|
| Ecoli     | PL-KNN        | 86.13 $\pm$ 0.45            |
|           | PL-LEAF       | 86.96 $\pm$ 0.83            |
|           | PL-ECOC       | 86.37 $\pm$ 0.75            |
|           | CLPL          | 85.00 $\pm$ 0.75            |
|           | PALOC         | 85.45 $\pm$ 0.86            |
|           | RegISL        | 80.39 $\pm$ 1.21            |
|           | PL-DNN+Loss1  | 89.56 $\pm$ 1.96            |
|           | PL-DNN+Loss2  | 89.43 $\pm$ 3.80            |
|           | PL-DNN+Loss3  | 90.74 $\pm$ 2.95            |
| Leaf      | PL-KNN        | 65.97 $\pm$ 1.17            |
|           | PL-LEAF       | 74.00 $\pm$ 1.52            |
|           | PL-ECOC       | 76.91 $\pm$ 1.50            |
|           | CLPL          | 54.18 $\pm$ 2.57            |
|           | PALOC         | 73.24 $\pm$ 1.86            |
|           | RegISL        | 70.85 $\pm$ 1.39            |
|           | PL-DNN+Loss1  | 87.07 $\pm$ 2.53            |
|           | PL-DNN+Loss2  | 92.00 $\pm$ 1.39            |
|           | PL-DNN+Loss3  | 90.22 $\pm$ 2.52            |
| Movement  | PL-KNN        | 76.25 $\pm$ 1.60            |
|           | PL-LEAF       | 82.06 $\pm$ 0.74            |
|           | PL-ECOC       | 84.97 $\pm$ 0.94            |
|           | CLPL          | 51.83 $\pm$ 2.27            |
|           | PALOC         | 84.83 $\pm$ 1.21            |
|           | RegISL        | 82.22 $\pm$ 0.98            |
|           | PL-DNN+Loss1  | 85.09 $\pm$ 2.55            |
|           | PL-DNN+Loss2  | 93.69 $\pm$ 1.06            |
|           | PL-DNN+Loss3  | 86.43 $\pm$ 7.60            |
| Vehicle   | PL-KNN        | 69.55 $\pm$ 0.70            |
|           | PL-LEAF       | 81.48 $\pm$ 0.56            |
|           | PL-ECOC       | 79.05 $\pm$ 1.51            |
|           | CLPL          | 75.39 $\pm$ 0.53            |
|           | PALOC         | 82.10 $\pm$ 0.78            |
|           | RegISL        | 70.61 $\pm$ 0.71            |
|           | PL-DNN+Loss1  | 80.68 $\pm$ 2.59            |
|           | PL-DNN+Loss2  | 82.41 $\pm$ 2.51            |
|           | PL-DNN+Loss3  | 81.64 $\pm$ 2.86            |
| CTG       | PL-KNN        | 74.67 $\pm$ 0.43            |
|           | PL-LEAF       | 80.94 $\pm$ 0.30            |
|           | PL-ECOC       | 83.03 $\pm$ 0.39            |
|           | CLPL          | 65.87 $\pm$ 0.24            |
|           | PALOC         | 82.96 $\pm$ 0.34            |
|           | RegISL        | 74.67 $\pm$ 0.76            |
|           | PL-DNN+Loss1  | 79.30 $\pm$ 1.69            |
|           | PL-DNN+Loss2  | 84.58 $\pm$ 1.45            |
|           | PL-DNN+Loss3  | 83.17 $\pm$ 1.11            |
| Segment   | PL-KNN        | 94.52 $\pm$ 0.21            |
|           | PL-LEAF       | 93.71 $\pm$ 0.10            |
|           | PL-ECOC       | 95.93 $\pm$ 0.24            |
|           | CLPL          | 82.73 $\pm$ 0.29            |
|           | PALOC         | 94.90 $\pm$ 0.16            |
|           | RegISL        | 95.61 $\pm$ 0.21            |
|           | PL-DNN+Loss1  | 91.82 $\pm$ 1.14            |
|           | PL-DNN+Loss2  | 95.86 $\pm$ 0.68            |
|           | PL-DNN+Loss3  | 95.84 $\pm$ 0.45            |
(a) Classification accuracy of PL-DNN algorithm when $lr$ is set to different values. (left: Lost, right: BirdSong)

(b) Classification accuracy of PL-DNN algorithm when train epoch is set to different values. (left: Lost, right: BirdSong)

(c) Classification accuracy of PL-DNN algorithm when $\lambda$ is set to different values. (left: Lost, right: BirdSong)

Fig. 2: Parameter sensitivity analysis for PL-DNN algorithm on the Lost and BirdSong datasets.
and 83% of the cases, the PL-DNN algorithm achieves superior performance against PL-KNN, PL-LEAF, PL-ECOC, CLPL, PALOC, and RegISL algorithms, respectively, and only on one controlled UCI dataset (Ecoli when $p=0.3$ and $r=1$) the RegISL algorithm outperforms PL-DNN algorithm.

D. Experimental results on the real-world partial label datasets

The performance of the proposed PL-DNN algorithm is also validated on five real-world partial label datasets named Lost [1], MSRCv2 [4], BirdSong [5], FG-NET [6] [7], and Soccer Player [3]. These datasets come from various real-world applications such as character-name association in TV show, ambiguous image classification, and image auto-annotation. The specific information of these datasets is presented in Table 1. Table 4 summarizes the classification accuracy of the PL-DNN algorithm and the other six algorithms on the real-world partial label datasets. In comparison with the six existing partial label learning algorithms, our deep neural network model combined with three newly defined loss functions exhibits better classification accuracy on the MSRCv2, FG-NET, and Soccer Player datasets, although the PL-LEAF, and CLPL algorithms can obtain the highest classification accuracy on the BirdSong and Lost datasets, respectively. Therefore, on the whole, the proposed PL-DNN algorithm can also achieve better accuracy on the real-world datasets.

To facilitate the reader to implement the partial label learning algorithm proposed in this study, we present the network structures for the real-world partial label datasets in Table 5. For example, $(108,90,60,30,16)$ indicates that the neural network includes one input layer, three hidden layers, and one output layer. The single input data is a row vector of length 108, the number of neurons in the hidden layer is 90, 60, and 30, and the number of neurons in the output layer is 16.

V. Conclusions

This study proposed a new partial label learning algorithm based on a deep neural network model. Experimental results on the controlled UCI datasets and real-world partial label datasets reveal that the deep neural network model combined with the newly defined loss functions could achieve better classification accuracy on most of the partial label datasets. The deep learning technology achieves good results on image and speech classification problems because it not only improves the ability of model representation by increasing the number of hidden layers but also integrates the procedures of feature extraction and classification [54]. In this study, we assume that the good features of each dataset have been extracted by a certain feature extraction technology. In the future work, we will build a partial label learning algorithm by using convolutional neural network model (CNN) in order to exploit the feature extraction ability of CNN for partial label learning problems.

Table 5: The DNN structures used on the real-world partial label datasets

| Data set  | Algorithms      | DNN structure               |
|-----------|-----------------|-----------------------------|
| Lost      | DNN+Loss1       | (108,90,80,70,60,50,30,16) |
|           | DNN+Loss2       | (108,150,100,50,16)        |
|           | DNN+Loss3       | (108,90,60,30,16)          |
| BirdSong  | DNN+Loss1       | (38,200,150,100,50,20,13) |
|           | DNN+Loss2       | (38,200,150,100,120,60,30,13) |
|           | DNN+Loss3       | (38,30,20,13)              |
| MSRCv2    | DNN+Loss1       | (48,300,200,100,50,23)    |
|           | DNN+Loss2       | (48,200,23)               |
|           | DNN+Loss3       | (48,200,100,23)           |
| FG-NET    | DNN+Loss1       | (262,300,256,200,100,78)  |
|           | DNN+Loss2       | (262,400,300,200,100,78)  |
|           | DNN+Loss3       | (262,100,78)              |
| Soccer-Player | DNN+Loss1    | (279,400,350,300,256,200,171) |
|           | DNN+Loss2       | (279,400,350,300,256,200,171) |
|           | DNN+Loss3       | (279,200,171)             |

ACKNOWLEDGMENT

This research was funded by the National Natural Science Foundation of China (62102062, 61972068, 62072152), the Natural Science Foundation of Liaoning Province (2020-MS-134, 2020-MZLH-29, 2018050625), and the Scientific Research Funding Project of the Educational Department of Liaoning Province (LJKZ0022).

REFERENCES

[1] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. Journal of Machine Learning Research, 12(May):1501–1536, 2011.
[2] J. Luo and F. Orabona. Learning from candidate labeling sets. In Advances in neural information processing systems, pages 1504–1512, 2010.
[3] Z. Zeng, S. Xiao, K. Jia, T.-H. Chan, S. Gao, D. Xu, and Y. Ma. Learning by associating ambiguously labeled images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 708–715, 2013.
[4] M.-L. Zhang, B.-B. Zhou, and X.-Y. Liu. Partial label learning via feature-aware disambiguation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1335–1344. ACM, 2016.
[5] E. Hüllermeier and J. Beringer. Learning from ambiguously labeled examples. Intelligent Data Analysis, 10(5):419–439, 2006.
[6] M.-L. Zhang and F. Yu. Solving the partial label learning problem: An instance-based approach. In Twenty-Fourth International Joint Conference on Artificial Intelligence, pages 4048–4054, 2015.
[7] N. Nguyen and R. Caruana. Classification with partial labels. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 551–559. ACM, 2008.
[8] F. Yu and M.-L. Zhang. Maximum margin partial label learning. Machine Learning, 106(4):573–593,
Table 3: Relative performance of PL-DNN algorithm against other algorithms on the controlled UCI datasets (Win/Tie/Loss).

| Data set       | PL-KNN   | PL-LEAF  | PL-ECOC | CLPL    | PALOC   | RegISL |
|----------------|----------|----------|---------|---------|---------|--------|
| Ecoli          | 4/0/0    | 3/1/0    | 4/0/0   | 4/0/0   | 3/1/0   | 3/0/1  |
| Leaf           | 4/0/0    | 4/0/0    | 4/0/0   | 4/0/0   | 4/0/0   | 4/0/0  |
| Movement       | 4/0/0    | 4/0/0    | 4/0/0   | 4/0/0   | 4/0/0   | 4/0/0  |
| Vehicle        | 4/0/0    | 2/2/0    | 3/1/0   | 4/0/0   | 3/1/0   | 4/0/0  |
| CTG            | 4/0/0    | 4/0/0    | 4/0/0   | 4/0/0   | 4/0/0   | 4/0/0  |
| Segment        | 4/0/0    | 1/3/0    | 4/0/0   | 4/0/0   | 1/3/0   | 4/0/0  |
| Total          | 24/0/0   | 21/3/0   | 20/4/0  | 24/0/0  | 22/2/0  | 20/3/1 |

Table 4: Classification accuracy of each algorithm on the real-world datasets(%).

| Algorithms          | PL-KNN   | PL-LEAF  | PL-ECOC | CLPL    | PALOC   | RegISL |
|---------------------|----------|----------|---------|---------|---------|--------|
|                     | Lost     | MSRCv2   | BirdSong | FG-NET | Soccer Player |
| PL-DNN+Loss1        | 36.42±0.50 | 43.86±0.42 | 65.55±0.30 | 4.24±0.33 | 49.36±0.07 |
| PL-DNN+Loss2        | 69.06±0.42 | 49.72±0.56 | 72.14±0.33 | 7.65±0.87 | N/A     |
| PL-DNN+Loss3        | 65.22±0.93 | 45.10±1.15 | 71.75±0.38 | 3.24±0.45 | N/A     |
| CLPL                | 69.15±0.71 | 40.89±0.73 | 63.88±0.16 | 7.65±0.69 | N/A     |
| PALOC               | 58.58±1.05 | 46.46±0.36 | 70.52±0.18 | N/A     | N/A     |
| RegISL              | 47.06±1.74 | 41.94±0.84 | 66.95±0.29 | 4.72±0.56 | N/A     |
| PL-DNN+Loss1        | 43.69±1.11 | 45.26±0.67 | 56.89±0.44 | 6.68±0.60 | 45.91±0.41 |
| PL-DNN+Loss2        | 56.97±2.12 | 50.82±0.58 | 54.74±0.91 | 7.81±0.77 | 50.88±0.20 |
| PL-DNN+Loss3        | 45.22±3.30 | 43.21±1.27 | 68.01±1.40 | 5.53±0.58 | 53.46±2.81 |

Apr 2017.
[9] Y. Grandvalet. Logistic regression for partial labels. In International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pages 1935–1941, 2002.
[10] A. Beygelzimer and J. Langford. The offset tree for learning with partial labels. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 129–138. ACM, 2009.
[11] F. Yu and M.-L. Zhang. Decision tree ensemble based partial label learning algorithm. Pattern Recognition and Artificial Intelligence, 29(4):367–375, 2016.
[12] D.-B. Wang, L. Li, and M.-L. Zhang. Adaptive graph guided disambiguation for partial label learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 83–91. ACM, 2019.
[13] F. Carrara, F. Falchi, R. Caldelli, G. Amato, and R. Becarelli. Adversarial image detection in deep neural networks. Multimedia Tools and Applications, 78(3):2815–2835, 2019.
[14] H. Seki, K. Yamamoto, T. Akiba, and S. Nakagawa. Discriminative learning of filterbank layer within deep neural network based speech recognition for speaker adaptation. IEICE Transactions on Information and Systems, 102(2):364–374, 2019.
[15] A. I. Harrouk and A. M. Barbar. A psycholinguistic approach to career selection using nlp with deep neural network classifiers. In IEEE International Multi-disciplinary Conference on Engineering Technology, pages 1–6. IEEE, 2018.
[16] X. Tang, H. Wang, and C. Jiao. Multi-objective evolutionary metric learning for image retrieval using convolutional neural network features. 2019 IEEE Congress on Evolutionary Computation, pages 578–585, 2019.
[17] X. Tang, C. Liu, J. Ma, X. Zhang, and L. Jiao. Large-scale remote sensing image retrieval based on semi-supervised adversarial hashing. Remote Sensing, 11(17):2055–, 2019.
[18] G. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. Advances in Neural Information Processing Systems, pages 2924–2932, 2014.
[19] N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: A tensor analysis. In Conference on Learning Theory, pages 698–728, 2016.
[20] A. Bentkamp, J. Blanchette, and D. Klakow. A formal proof of the expressiveness of deep learning. Journal of Automated Reasoning, 63(2):347–368, Aug. 2019.
[21] S. Sun, W. Chen, L. Wang, X. Liu, and T. Liu. On the depth of deep neural networks: A theoretical view. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016.
[22] L. Liu and T. Dietterich. Learnability of the superset label learning problem. In International Conference on Machine Learning, pages 1629–1637. PMLR, 2014.
[23] E. Hullemeier and W. Cheng. Superset learning based on generalized loss minimization. In Joint Eu-
European Conference on Machine Learning and Knowledge Discovery in Databases, 2015.

[24] Y. Chen, V. M. Patel, R. Chellappa, and P. J. Phillips. Ambiguously labeled learning using dictionaries. *IEEE Transactions on Information Forensics and Security*, 9(12):2076–2088, 2014.

[25] C. H. Chen, V. M. Patel, and R. Chellappa. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis & Machine Intelligence, PP*(99):1–1, 2017.

[26] G. Lyu, S. Feng, T. Wang, C. Lang, and Y. Li. Gm-pll: Graph matching based partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(2):521–535, 2021.

[27] L. Feng and B. An. Partial label learning with self-guided retraining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3542–3549, 2019.

[28] N. Xu, J. Lv, and X. Geng. Partial label learning via label enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5557–5564, 2019.

[29] G. Lyu, S. Feng, Y. Jin, and Y. Li. Partial label learning via self-paced curriculum strategy. In *Machine Learning and Knowledge Discovery in Databases: European Conference*, pages 489–505. Springer International Publishing, 2021.

[30] G. Lyu, S. Feng, T. Wang, and C. Lang. A self-paced regularization framework for partial-label learning. *IEEE Transactions on Cybernetics*, 2020.

[31] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning*, pages 6500–6510. PMLR, 2020.

[32] Y. Yao, C. Gong, J. Deng, and J. Yang. Network cooperation with progressive disambiguation for partial label learning. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2020.

[33] F. Lei and A. Bo. Leveraging latent label distributions for partial label learning. In *Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018.

[34] C. Gong, T. Liu, Y. Tang, J. Yang, J. Yang, and D. Tao. A regularization approach for instance-based superset label learning. *IEEE transactions on cybernetics*, 48(3):967–978, 2017.

[35] X. Wu and M.-L. Zhang. Towards enabling binary decomposition for partial label learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2868–2874, 2018.

[36] L. Feng and B. An. Partial label learning by semantic difference maximization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2294–2300. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[37] Y. Yan and Y. Guo. Partial label learning with batch label correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6575–6582, 2020.

[38] J. Wang and M.-L. Zhang. Towards mitigating the class-imbalance problem for partial label learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[39] M.-K. Xie and S.-J. Huang. Partial multi-label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[40] Y. Yan and Y. Guo. Adversarial partial multi-label learning. *ArXiv*, abs/1909.06717, 2019.

[41] Y. Zhou and H. Gu. Geometric mean metric learning for partial label data. *Neurocomputing*, 275:394–402, 2018.

[42] S. Xu, M. Yang, Y. Zhou, R. Zheng, W. Liu, and J. He. Partial label metric learning by collapsing classes. *International Journal of Machine Learning and Cybernetics*, 11:2453–2460, 2020.

[43] Y. Zhou, J. He, and H. Gu. Partial label learning via gaussian processes. *IEEE Transactions on Cybernetics*, 47(12):4443–4450, Dec 2017.

[44] L. Xing-si. An aggregate function method for nonlinear programming. *Science in China Series A-Mathematics, Physics, Astronomy & Technological Science*, 34(12):1467–1473, 1991.

[45] G. Shi, J. Zhang, H. Li, and C. Wang. Enhance the performance of deep neural networks via l2 regularization on the input of activations. *Neural Processing Letters*, pages 1–19, 2018.

[46] Q. Zheng, J. Fang, Z. Hu, and H. Zhang. Aero-engine on-board model based on batch normalize deep neural network. *IEEE Access*, 7:54855–54862, 2019.

[47] M.-L. Zhang, F. Yu, and C.-Z. Tang. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2155–2167, 2017.

[48] L. Liu and T. G. Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in neural information processing systems*, pages 548–556, 2012.

[49] F. Briggs, X. Z. Fern, and R. Raich. Rank-loss support instance machines for miml instance annotation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 534–542. ACM, 2012.

[50] H. Han, C. Otto, X. Liu, and A. K. Jain. Demographic estimation from face images: Human vs. machine performance. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1148–1161, 2014.

[51] A. Lanitis, C. J. Taylor, and T. F. Cootes. Toward automatic simulation of aging effects on face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):442–455, 2002.

[52] A. Errachdi, M. Benrejeb Model reference adaptive control based-on neural networks for nonlinear
time-varying system. *International Journal of Applied Mathematics, Computational Science and Systems Engineering*, 1: 6-10, 2019.

[53] N. Rouzegari, V. Nourani, A. Molajou Application of artificial neural network and predictor screening method for downscaling climatic parameters, *International Journal of Environmental Science*, 4: 112-119, 2019.

[54] A. Khan, A. Sohail, U. Zahoora, A. S. Qureshi A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53: 5455-5516, 2020.

Creative Commons Attribution License 4.0 (Attribution 4.0 International , CC BY 4.0)
This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US