Knowledge-Based Systems 245 (2022) 108602

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Deep semi-supervised learning with contrastive learning and partial label propagation for image data

Yanglan Gan [a], Huichun Zhu [a], Wenjing Guo [a], Guangwei Xu [a], Guobing Zou [b,*]

[a] School of Computer Science and Technology, Donghua University, Shanghai, 201620, China
[b] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

ARTICLE INFO

Article history:
Received 5 November 2021
Received in revised form 13 March 2022
Accepted 14 March 2022
Available online 21 March 2022

Keywords:
Semi-supervised learning
Contrastive learning
Partial label propagation
Data augmentation

ABSTRACT

Deep semi-supervised learning is becoming an active research topic because it jointly utilizes labeled and unlabeled samples in training deep neural networks. Recent advances are mainly focused on inductive semi-supervised learning which generally extends supervised algorithms to include unlabeled data. In this paper, we propose CL_PLP, a new transductive deep semi-supervised learning algorithm based on contrastive self-supervised learning and partial label propagation. The proposed method consists of two modules, contrastive self-supervised learning module extracting features from labeled and unlabeled data and partial label propagation module generating confident pseudo-labels through label propagation. For contrastive learning, we propose an improved twins network model by adding multiple projector layers and the contrastive loss term. Meanwhile, we adopt strong and weak data augmentation to increase the diversity of the dataset and the robustness of the model. For the partial label propagation module, we interrupt the label propagation process according to the quality of pseudo-labels and improve the impact of high-quality pseudo-labels. The performance of our algorithm on three standard baseline datasets CIFAR-10, CIFAR-100 and miniImageNet is better than previous state-of-the-art transductive deep semi-supervised learning methods. By transferring our model to the medical COVID19-Xray dataset, it also achieves good performance. Finally, we propose a strategy to integrate our partial label propagation module with inductive semi-supervised learning method, and the results prove that it can further improve their performance and obtain additional high-quality pseudo-labels for the unlabeled data.

1. Introduction

In recent years, deep learning has shown remarkable successes in various fields, which partially benefits from large numbers of labeled data used to train deep neural networks [1]. However, as the process of annotating labels such as artificial annotation for the data is usually time-consuming and expensive [2], building large labeled datasets for different deep learning tasks is practically unfeasible. Therefore, it is an important research topic to develop semi-supervised deep learning approaches that can jointly learn from labeled and unlabeled samples [3].

Semi-supervised learning methods can be divided into two main categories, including inductive and transductive learning [4]. Inductive semi-supervised learning trains a model on the input dataset in various ways, and then applies the pre-trained model to generate predictions for unseen samples. Differently, transductive semi-supervised learning does not construct a classifier for the entire input dataset. Their predictive power is only limited to the data encountered during the training phase. Previous deep semi-supervised learning algorithms are mainly built on the inductive semi-supervised learning framework [5,6], whereas classical transductive idea and manifold assumption is rarely exploited. Recently, the Label Propagation for Deep Semi-supervised Learning (DLP) has been proposed [7], which adopts transductive label propagation strategy to infer additional pseudo-labels for unlabeled data, and further utilizes the generated pseudo-labels to train the classifier. Specifically, pseudo-labeling methods utilize the labeled data model to predict labels for unlabeled data. As those predicted labels may or may not be real labels, they are regarded as pseudo-labels. However, during the feature representation procedure, DLP just utilizes the information of labeled data, rather than making full use of the unlabeled data, which might lead to the poor quality of pseudo-labels and affect its performance.

To avoid time-consuming and expensive data annotations, self-supervised learning methods are also proposed to learn features from large-scale unlabeled data [1]. Self-supervised learning is a kind of unsupervised learning approach, which essentially

* Corresponding author.
E-mail addresses: ylgan@dhu.edu.cn (Y. Gan), 2202540@mail.dhu.edu.cn (H. Zhu), wjguo0@dhu.edu.cn (W. Guo), gwxu@dhu.edu.cn (G. Xu), gbzou@shu.edu.cn (G. Zou).

https://doi.org/10.1016/j.knosys.2022.108602
0950-7051/

does not need any label information, and can extract useful information from the data itself. To learn an effective feature representation from the unlabeled data, a widely-used solution is to design various pretext tasks for pre-training the network structure [8,9]. By optimizing objective functions of the pretext tasks, the networks can be trained and the feature representation can be learned. Subsequently, a few labels are utilized to further refine the pre-trained network and the pre-trained network is applied to specific downstream tasks [10,11].

In this paper, we propose a new deep semi-supervised learning algorithm based on contrastive self-supervised learning and partial label propagation strategy, called CL_PLP. The proposed method consists of two modules, including a self-supervised feature extraction module and a partial label propagation module, which can respectively improve two stages of the traditional label propagation methods. When the labels are insufficient, it is difficult to learn accurate feature representation by a trained network, which further decreases the accuracy of pseudo-labels generated from the label propagation stage. To tackle this problem, we propose a new network structure, adding projection layers and an additional contrastive loss term for contrastive learning. Meanwhile, we expand the dataset by combining strong and weak augmentation [6] to increase the diversity of the dataset and the robustness of the model. For the second stage, we improve the impact of high-confidence pseudo-labels by interrupting the label propagation procedure according to the quality of pseudo-labels. Finally, we propose a strategy to integrate our partial label propagation module with state-of-the-art inductive semi-supervised learning algorithms.

On the standard datasets CIFAR-10, CIFAR-100 and miniImageNet, the experimental results demonstrate that the performance of our algorithm is superior to previous state-of-the-art deep semi-supervised algorithms, especially when the labeled data are fewer, the improvement effect is more obvious. Moreover, we apply our algorithm to a medical image dataset COVID19-Xray, which also achieves good performance.

The main contributions of this paper are summarized as follows:

- We propose a new transductive deep semi-supervised learning algorithm based on contrastive self-supervised learning and partial label propagation strategy. Compared to previous transductive methods, the advantage lies in that it adopts contrastive learning to extract features from labeled and unlabeled data, and utilizes partial label propagation strategy to interrupt label propagation and obtain highly confident pseudo-labels.
- Based on the proposed method, we further introduce an improvement strategy for inductive semi-supervised learning methods. By integrating our partial label propagation module, inductive semi-supervised learning methods can obtain high-quality pseudo-labels of unlabeled data and consequently improve their performance.
- Extensive experimental results demonstrate the effectiveness and applicability of our algorithm. On the standard datasets and medical COVID19-Xray dataset, our algorithm outperforms the compared methods. Meanwhile, the experimental results demonstrate that the integration of our proposed method with inductive semi-supervised learning methods achieves better performance.

The remainder of the paper is organized as follows. Related work is introduced in the following section, and each step of our algorithm is described in detail in Section 3. Section 4 includes our experimental results and ablation study. Finally, we conclude in Section 5 with a summary and an outlook on future work.

2. Related work

The fundamental purpose of semi-supervised and self-supervised learning is to exploit large numbers of unlabeled data to obtain useful information for specific downstream tasks [1,2]. This section introduces the most relevant methods of utilizing unlabeled data in these fields, which our algorithm is built on.

2.1. Consistency loss and entropy minimization

Consistency loss. For semi-supervised learning algorithms, the common strategy of utilizing unlabeled data is to construct unsupervised consistency loss, which can convert the existing supervised learning algorithm into semi-supervised learning. In the Ladder network [12], Π-model [13] and other semi-supervised learning algorithms, the data or the networks are randomly perturbed, and the unsupervised consistency loss is calculated according to the output before and after perturbation [14–16]. Accordingly, the consistency loss is defined as:

$$\mathcal{L}(X) = \|f(X_1;\theta_1) - f(X_2;\theta_2)\|^2 \tag{1}$$

where X_1 and X_2 are two augmented versions of the whole dataset X, θ_1 and θ_2 are the parameters of two noised version model, $\|\cdot\|$ represents L2 norm, $f(\cdot)$ corresponds to the output of the network.

Entropy minimization. Entropy minimization is a widely used regularization technology [7,17]. In order to obtain better outputs of network, we can constrain the entropy of the prediction outputs to be minimized. There are many types of information entropy, including KL divergence and cross entropy [18]. Specifically, KL divergence is used to measure the difference between two probability distributions, and cross entropy is utilized to evaluate the closeness between the actual output and the expected output. When the data label is converted into a one-hot vector, the above two types of entropy can be easily used as the loss function. In semi-supervised learning, entropy minimization regularization is often adopted to optimize the network prediction of unlabeled data and to minimize its uncertainty [5,19]. The two classical entropy minimization can be respectively formalized as:

$$\min CE\big(f(X_L;\theta), Y_L\big) = -\sum_{i=1}^{l} y_i \log f(X_i;\theta) \tag{2}$$

$$\min E\big(f(X_U;\theta)\big) = -\sum_{i=l+1}^{n} f(X_i;\theta)\log f(X_i;\theta) \tag{3}$$

where X_L and X_U denotes labeled data and unlabeled data, θ denotes the parameters of network, CE and E denotes CrossEntropy and Entropy respectively.

2.2. Data augmentation

Data augmentation has been widely used in both supervised and unsupervised representation learning. It is an intuitive way to expand the number of data. When labeled data are insufficient, the most direct approach is to expand the labeled dataset. The traditional methods for expanding are oversampling and random perturbation. However, in deep learning, only oversampling data often makes the network fall into the trap of overfitting. Therefore, the data expansion for deep learning should be expanded in both quantity and diversity [20]. SMOTE [21] is a well-known oversampling algorithm, which generates synthetic samples, but it does not consider the diversity of generated data. The variations of SMOTE, such as Safe-Level SMOTE [22] and SMOTE_RSB [12], improve the diversity of the expanded data. Generative adversarial networks can also be used for data augmentation. But

in the field of computer vision, the classical approach for data expansion is to transform the image, that is, image augmentation, including random rotation, random translation, random clipping, random color distortion, and so on [1,11]. In FixMatch [6], image augmentation is firstly divided into strong and weak augmentation. Strong augmentation refers to the image augmentation that changes the pixel value of the image randomly. On the contrary, weak augmentation does not change the pixel value of the image, that is, random clip or flip the whole image.

2.3. Self-supervised contrastive learning

Self-Supervised learning is another important way to utilize unlabeled data. The key for self-supervised learning is designing proper pretext tasks and utilizing data itself to train models [1]. Gidaris et al. designed a pretext task based on image rotation prediction [23], trained the network to predict the rotation degree and then fine-tuned on labeled data for image classification. The S^4L [11] algorithm designs a class of pretext tasks that are independent of specific downstream tasks, and then converts unsupervised learning into supervised learning. The pretext tasks might be to predict the rotation degree or the transformation mode of the image. Subsequently, the network trained in the previous stage is fine-tuned with a few labeled data to get the final network.

Contrastive learning is a powerful approach in the field of self-supervised learning, whose main idea is to learn representations so that similar samples stay close to each other, while dissimilar ones are far apart. SimCLR [8] regards contrastive learning as the pretext task. It inputs two versions of randomly noised data into two similar networks respectively, compares the output of two networks, and utilizes a self-supervised contrastive loss to train the network [9]. In this work, they view the different augmented version for the given sample in a big batch as the positive pair, and view the different noised version of different samples as the negative pairs. MoCo [10] and MoCoV2 [24] also think the negative pairs of samples are important for contrastive learning. They explicitly maintain a queue of negative examples from the past mini-batches. The above algorithms either need to maintain additional queues to store negative pairs, or need to use a larger batch_size to get as many negative pairs as possible. Differently, BYOL [25] and SimSiam [26] are proposed to obtain better parameters of network using only positive pair sample. In their works, by stopping gradient one network of two similar networks, the loss of negative pair can be avoided.

2.4. Label propagation for deep semi-supervised learning

Label propagation has been gradually used in transductive learning. The method TPLPA [27] calculates the topological potential of each node and consider it as the importance of the node. Then it sorts nodes according to the importance and selects the most important node in the propagation. kNN-LDP [28] employs a Bayesian schema to propagate the label probability distribution to neighbors, rather than propagating an immediate label decision. MTL-SSLP [29] combines multi-task learning with semi-supervised label propagation strategy. It uses a probabilistic map to guide a semi-supervised label propagation process. NLPPC [30] proposes a new version of label propagation, which exploits pairwise relations of labels as constraints to construct an optimization model for propagating labels.

The method DLP [7] utilizes transductive label propagation to infer pseudo-labels [31] for unlabeled data, which are further used to train the classifier. The algorithm consists of two stages. In the first stage, only labeled data are used to pretrain a network which will be used to extract the feature representation of all data. In the second stage, the adjacent graph is constructed based on the feature representation extracted, and the label propagation is carried out to obtain the pseudo-labels of unlabeled data. Subsequently, the pseudo-labeled and labeled data are returned to the first stage, and the two steps start to iterate. It is worth noting that the unlabeled data initially are not used in the first stage, which might lead to the following situation. When the labeled data is insufficient, training data might lack representativeness, which makes the network fall into overfitting problem. Then the first stage is hard to train a network with a set of superior parameters that can extract effective features. Although the authors realized this problem and integrated the label propagation process with the MT algorithm, due to its weak data augmentation methods, data representativeness is still insufficient, and the overfitting problem is unsolved during training the model. Meanwhile, the quality of pseudo labels obtained in the second stage is critical to the subsequent process of training. If the quality is poor, the performance might become worse with the process iterates, leading to an "avalanche phenomenon". In this paper, we propose several strategies to overcome these limitations, which will be elaborated in the following method section.

3. Proposed method

We propose CL_PLP, a new deep transductive semi-supervised algorithm based on contrastive self-supervised learning and partial label propagation strategy. This method is divided into two steps. In the first step, we first perform data augmentation by randomly combining strong and weak augmentation. Based on all augmented data (including labeled and unlabeled data), we train twin networks and optimize the network parameters through symmetric contrastive loss, supervised and unsupervised loss. Specifically, twin networks are two networks with exactly the same network structure. One of the sub-networks does not optimize the network parameters by gradient descent, whose parameters are obtained by average exponential moving (EMA) [32] the network parameters of the other sub-network. In the second step, the pre-trained network is utilized to extract the feature representation of the whole dataset. Next, the k-nearest neighbor graph is constructed based on the feature representation, and the pseudo-labels of unlabeled data will be inferred by partial label propagation. During this process, we set a confidence threshold to interrupt the label propagation which may generate low-quality pseudo-labels. After partial label propagation, we obtain three types of data, including originally labeled data, pseudo-labeled data and unlabeled data, which would return to the network to further optimize its parameters. Then our learning process iterates between these two steps. The framework of our algorithm is illustrated in Fig. 1. The details will be elaborated in the following sections.

3.1. Formalization

Our original inputs include labeled data and unlabeled data, and $X = \{x_1, x_2, \ldots, x_l, x_{l+1}, \ldots, x_n\}$ denotes the whole dataset. $X_L = \{x_1, x_2, \ldots, x_l\}$ and $X_U = \{x_{l+1}, x_{l+2}, \ldots, x_n\}$ respectively denotes the labeled and unlabeled dataset. Each sample x_i in the labeled dataset X_L has a definite label $y_i \in Y_L = \{1, 2, \ldots, c\}$. The problems can be formally described as: use the whole dataset X to train a deep network with better parameters, and annotate the known unlabeled data X_U with high quality pseudo-labels $\tilde{Y}_U = \{\tilde{y}_{l+1}, \tilde{y}_{l+2}, \ldots, \tilde{y}_n\}$. Meanwhile, we can generate the prediction for unseen data by using the trained network.

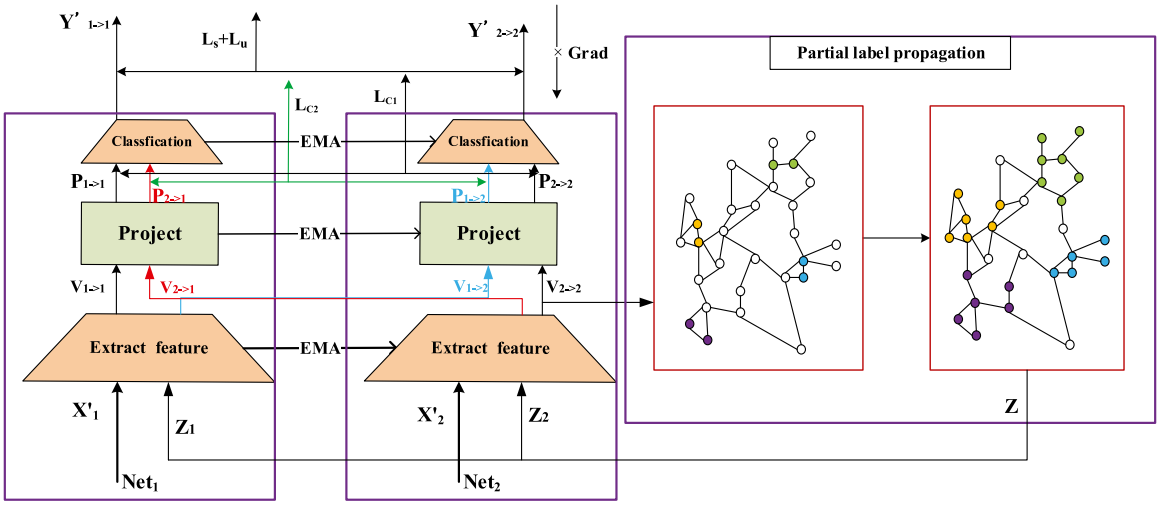

Fig. 1. The framework of our algorithm. Net_1 and Net_2 respectively represents two sub-networks. The direction of the arrow represents the flow of data. Among them, $V_{1\to1}$ represents the feature representation flow from Net_1 to Net_1. Similarly, $V_{1\to2}$ represent the feature representation flowing from Net_1 to Net_2, and $P_{1\to1}$ and $Y'_{1\to1}$ respectively are the projections and predictions of $V_{1\to1}$. The partial label propagation part on the right is the core part of the second step. In the graph, each color represents a class. In the initial state, each class has only a few labeled samples (colored nodes). After the partial label propagation process, the node is newly stained with color, and the white node represents the data with lower quality of pseudo-labels filtered by the threshold, which is still regarded as unlabeled data, and the pseudo-labels are discarded. Then three types of data can be obtained: originally labeled data, unlabeled data, and pseudo-labeled data.

3.2. Contrastive learning

3.2.1. Model structure

Our model consists of two networks with the same structure called symmetric twin networks. As shown in Fig. 1, Net_1 and Net_2 respectively represents two sub-networks, which respectively consists of three modules, including a feature extraction module, a projector module and a classifier module. Data streams across the model can be described as below. X'_1 and X'_2 are two augmented versions of X, obviously the labels of X'_L are still Y_L. Take the forward process of the sub-network Net_1 as an example, X'_1 is firstly input into the feature extraction module, and then this module outputs the feature representation $V_{1\to1}$ and $V_{1\to2}$ (transferred to the other sub-network) of X'_1. Subsequently, the feature $V_{1\to1}$ continues to forward to the projector module, to obtain the $P_{1\to1}$. Finally, $V_{1\to1}$ inputs into the classifier module, and the classifier module outputs the prediction $Y'_{1\to1}$ of X'_1 generated by Net_1.

3.2.2. Loss function

Supervised loss is designed for labeled data, which is usually utilized in supervised learning algorithms to train a deep neural network by minimizing the loss [19,33,34]. In our algorithm, we also construct a supervised loss as:

$$\mathcal{L}_S\left(X'_L, Y_L; \theta\right) = \sum_{i=1}^{l} \mathcal{L}_s\left(f(x'_i; \theta), y_i\right) \tag{4}$$

where X'_L denotes the augmentative version of X_L, θ denotes the parameter set of the networks, $f(x'_i; \theta)$ is the output of our network, and $\mathcal{L}_S$ represents the loss function for supervised learning, whose classical choice is cross-entropy, see Eq. (2).

Unsupervise loss is designed for both labeled and unlabeled data, which is critical to semi-supervised learning algorithms. Previous studies also name this loss term as a consistency loss. It is utilized to evaluate the consistency of two outputs corresponding to two versions of the input data [35], including before and after perturbation or augmentation. Here, we simultaneously introduce data and network perturbation. For data perturbation, we randomly combine strong and weak data augmentation. For network perturbation, we adopt average exponential moving (EMA) of network parameters. The unsupervised loss in our algorithm is defined as:

$$\mathcal{L}_U\left(X'; \theta\right) = \sum_{i=1}^{n} \mathcal{L}_u\left(f(x'_{1,i}; \theta_1), f(x'_{2,i}; \theta_2)\right) \tag{5}$$

where $\mathcal{L}_u$ denotes the unsupervised loss function, and θ_i denotes the parameter of the ith sub-network. The widely-used loss functions are MSE or KL-divergence [19,32], MSE can be calculated as $\frac{1}{m}\sum_{i=1}^{m}\left(f(x'_{1,i}; \theta_1) - f(x'_{2,i}; \theta_2)\right)^2$, and the KL-divergence is computed as $\mathrm{KL}(p \parallel q) = \sum p(x) \log \frac{p(x)}{q(x)}$.

Symmetrical Contrastive Loss. The contrastive loss is actually derived from the concept of self-supervised contrastive learning, which is used to judge whether two samples are similar. Contrastive loss considers the loss of positive pairs and negative pairs of samples. Here, we define the symmetrical contrastive loss between positive pairs as:

$$\mathcal{L}_C\left(X'; \theta\right) = \alpha\mathcal{L}_{C1} + \beta\mathcal{L}_{C2} \tag{6}$$

$$\mathcal{L}_{C1}\left(X'; \theta\right) = \sum_{i=1}^{n} \mathcal{L}_c\left(P_{1\to1,i}, P_{2\to2,i}\right) \tag{7}$$

$$\mathcal{L}_{C2}(X'; \theta) = \sum_{i=1}^{n} \mathcal{L}_c\left(P_{1\to2,i}, P_{2\to1,i}\right) \tag{8}$$

where α and β are the proportion of two contrastive losses, whose sum commonly equals 1 and the default values are 0.5. $P_{n1\to n2,i}$ denotes the output of ith sample in projector module, $n1$ and $n2$ refer to the $n1$-th augmented version and the $n2$-th sub-network respectively.

3.2.3. Joint optimization

The loss function for the proposed model consists of three loss terms: a supervised loss, an unsupervised loss and a symmetrical contrastive loss. $\mathcal{L}_S$ measures the difference between network predictions and true labels. $\mathcal{L}_U$ depends on the smoothness assumption, that is, for two inputs x_1 and x_2 that are the two augmented versions of x, the corresponding outputs y'_1 and y'_2 should be the similar, see Eq. (5). $\mathcal{L}_C$ measures the difference between two features extracted by projector layers of positive

pairs of samples. In the first stage of training, we iterate T epochs to jointly optimize these three kinds of loss functions. The total loss is calculated as:

$$\mathcal{L}_1 = \mathcal{L}_S + \mathcal{L}_U + \mathcal{L}_C \tag{9}$$

where the supervised loss $\mathcal{L}_S$ is only for the initial available labeled data in the dataset, which is different from the overall loss in the second stage.

3.3. Partial label propagation

After the first stage, we obtain a trained deep network with better network parameters called the pre-trained model. We further utilize the pre-trained model in the second stage.

Construction of kNN graph. We input the entire data X into the pre-trained model to extract the features $V = (v_1, \ldots, v_l, v_{l+1}, \ldots, v_n)$, where $v_i = f_{extract}(x_i; \theta_{extract})$, where $f_{extract}(\cdot)$ and $\theta_{extract}$ denotes the output and parameters of our feature extraction module, and then use those features to construct the similarity k adjacency graph of the data. In this step, we will first construct a sparse affinity matrix $A \in \mathbb{R}^{n\times n}$, whose elements are calculated as:

$$a_{ij} = \begin{cases} \left[v_i^T \cdot v_j\right], & \text{if } i \neq j \wedge v_i \in \mathrm{NN_k}(v_j) \\ 0 \quad , & otherwise \end{cases} \tag{10}$$

where $\mathrm{NN_k}(v_j)$ denotes the set of k neatest neighbors of v_j in X. Then we utilize the affinity matrix A to construct the weight matrix W_0 of the neighboring graph as $W_0 = A + A^T$, and the normalized counterpart calculated by $W = D^{-1/2}W_0D^{-1/2}$, where D is the degree matrix of W_0.

Label propagation [4]. We construct a label matrix $M \in \mathbb{R}^{n\times c}$ with initialized elements as:

$$M_{ij} = \begin{cases} 1, & \text{if } i \in \left[1, l\right] \wedge y_i = j \\ 0, & otherwise \end{cases} \tag{11}$$

As the previous study [4] did, we propagate label information from X_L to X_U in feature space. The label propagation rule is to propagate label information in an iterative manner, as shown in Eq. (12).

$$Z^t = \mu WZ^{t-1} + (1-\mu)M \tag{12}$$

where $\mu \in [0, 1)$ is the trade-off hyperparameter for WZ^t and M, and $Z^0 = M$. Eq. (12) iterates to convergence, which can be obtained:

$$Z^* = \lim_{t\to\infty} Z^t = (1-\mu)(I-\mu W)^{-1}M \tag{13}$$

Proof Eq. (13). According Eq. (12), we can obtain:

$$\begin{aligned} Z^t &= \mu W[\mu WZ^{t-2} + (1-\mu)M] + (1-\mu)M \\ &= \mu W\{\mu W[\mu WZ^{t-3} + (1-\mu)M] + (1-\mu)M\} \\ &\quad + (1-\mu)M \\ &= \mu W\{\mu W[\mu W\ldots + (1-\mu)M] + (1-\mu)M\} \\ &\quad + (1-\mu)M \\ &= (\mu W)^t M + (1-\mu)\left(\sum_{i=0}^{t-1}(\mu W)^i\right)M \end{aligned} \tag{14}$$

Thus, the limitation of Z can be defined as follows:

$$Z^* = \lim_{t\to\infty} Z^t = \lim_{t\to\infty}(\mu W)^t M + \lim_{t\to\infty}(1-\mu)\left(\sum_{i=0}^{t-1}(\mu W)^i\right)M \tag{15}$$

As for $\lim_{t\to\infty}(\mu W)^t$, the normalized symmetric Laplacian matrix L_{sys} of the graph can be given as:

$$L_{sys} = I - D^{-1/2}W_0D^{-1/2} = I - W \tag{16}$$

In fact, the eigenvalues λ of the normalized symmetric Laplacian satisfy $0 = \lambda_0 \leq \cdots \leq \lambda_{n-1} \leq 2$ [36]. Therefore, the eigenvalues λ' of the W satisfy $-1 = \lambda'_0 \leq \cdots \leq \lambda'_{n-1} \leq 1$, and $\mu \in [0, 1)$, we can obtain:

$$\lim_{t\to\infty}(\mu W)^t = 0 \tag{17}$$

In fact, $\sum_{i=0}^{t-1}(\mu W)^i$ is the sum of geometric progression, so that,

$$\begin{aligned} \lim_{t\to\infty}\left(\sum_{i=0}^{t-1}(\mu W)^i\right) &= \frac{I - \lim_{t\to\infty}(\mu W)^t}{I-\mu W} = \frac{I}{I-\mu W} \\ &= (I-\mu W)^{-1} \end{aligned} \tag{18}$$

Then we can compute a diffused matrix Z' by using the conjugate gradient (CG) [7,37] method to solve the following linear equation:

$$\left(I-\mu W\right)Z' = M \tag{19}$$

where $\mu \in [0, 1)$ is a hyperparameter. I is a identity matrix. After we obtain the matrix Z', we can infer the pseudo-labels of unlabeled data by the following formula:

$$y'_i = \arg\max_j z'_{ij} \tag{20}$$

Pseudo-label filtering and class-imbalanc constraints. Here, by normalizing Z', we can achieve soft assignments for classification. Each row of Z' represents the probability that a sample belongs to different classes. Since pseudo-labeling may produce false pseudo-labeled samples, which would reversely mislead the model with wrong information of labels and learn the incorrect feature representation. Therefore, we set a label propagation threshold η to interrupt the unconfident label propagation process and discard propagation with low confidence, which is formalized as:

$$y'_i = \begin{cases} \arg\max_j z'_{ij}, & \text{if } entropy(\tilde{z'}_i) < \eta \wedge i \in \left[l+1, n\right] \\ -1, & \text{if } entropy(\tilde{z'}_i) > \eta \wedge i \in \left[l+1, n\right] \\ y_i, & \text{if } i \in \left[1, l\right] \end{cases} \tag{21}$$

Meanwhile, we can calculate the information entropy as the confidence of the pseudo-label for this sample. Smaller information entropy indicates higher quality of the pseudo label. Then, each sample is assigned a weight ω to indicate the quality of the inferred pseudo-label, which is further used to weight the supervised loss of the sample in subsequent iterative training process. Here, we assign the ω according to different types of data that have been partially propagated, which is defined as:

$$\omega_i = \begin{cases} 1 - \frac{entropy(\tilde{z'}_i)}{\log(c)}, & \text{if } entropy(\tilde{z'}_i) < \eta \wedge i \in \left[l+1, n\right] \\ 0, & \text{if } entropy(\tilde{z'}_i) > \eta \wedge i \in \left[l+1, n\right] \end{cases} \tag{22}$$

where $i \in (l, n]$, the $\tilde{z'}_i$ denotes the ith row of normalized Z', and c is the number of classes.

When we obtain pseudo-labels through partial label propagation step, class-imbalanced situation often appears. To tackle this problem, we count the number of samples belonging to different classes, and assign each category a weight with $\xi_j = \left(|L_j|+|U_j|\right)^{-1}$, where L_j and U_j represent labeled and pseudo-labeled samples belong to j class respectively, and $|\cdot|$ represents the number of samples.

3.4. Iterative training

Based on these two steps, we obtain three kinds of data, including labeled, pseudo-labeled, and unlabeled data. Here, to increase the diversity of the data, we resample the labeled data

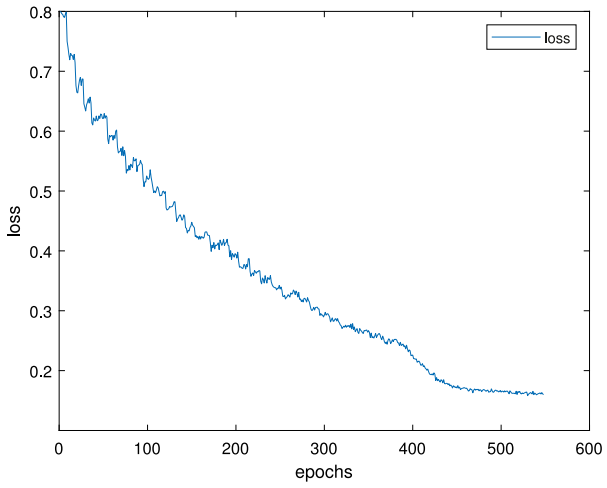

Fig. 2. The loss of iterative training according to ground-truth on CIFAR-10 with 1000 labeled images.

and high-confident pseudo-labeled data selected by previous partial label propagation process, which can be further processed with strong augmentations. These data will return to the pre-trained deep network in the first stage, and the two stages are iterated multiple rounds to fine-tune the network parameters by optimizing the loss defined:

$$\mathcal{L}_{all} = \omega\xi\mathcal{L}_{S'} + \mathcal{L}_U + \mathcal{L}_C \tag{23}$$

$$\begin{aligned}\mathcal{L}_{S'}(X, Y_L, Y'_U; \theta) = &\sum_{i=1}^{l} \mathcal{L}_{s'}\big(f(x_i;\theta), y_i\big) \\ &+ \sum_{i=l+1}^{n} \mathcal{L}_{s'}\big(f(x_i;\theta), y'_i\big)\end{aligned} \tag{24}$$

Finally, we can obtain a trained network, which can be applied to predict the labels for unseen data, and to obtain the pseudo-labels of unlabeled data for training through label propagation. The latter is the critical difference from the traditional inductive semi-supervised algorithm.

Further, as the proposed method conducts iterating training based on the two major modules, we investigate its convergence from the following two aspects. For the partial label propagation module, the convergence is proved in previous subsection (Partial Label Propagation). For the deep neural network module, as the neural network is over-parameterized and structurally symmetric, there will be a large number of equivalent solutions, and the final result is highly dependent on the initial conditions. In fact, the goal of deep learning is to obtain the smallest generalization error, not necessarily converging to the lowest point. The convergence of the neural network can be reflected by the decreasing curve of the loss with the training epochs. Therefore, we train 540 epochs on the CIFAR-10 using 100 labeled data from each class, and plot the decreasing curve of the loss in Fig. 2. From the curve, we can observe that the loss will eventually converge to a smaller value.

4. Experiment and analysis

In this section, we present the results of our main experiments. We first focus on three standard baseline datasets CIFAR-10, CIFAR-100 and miniImageNet. Further, to evaluate the domain adaptability, we transfer the proposed method to the biomedical image dataset COVID19-Xray.

Algorithm 1 CL_PLP

Input: $X = \{X_L, X_U\}$; Y_L
Output: Y'_U;Network with better parameters θ

```
θ ← initialize randomly
for epoch ∈ [1, ..., T] do
    resample X_L and augment the whole data X
    L_1 = L_S + L_U + L_C
    θ ←OPTIMIZE(L_1)
end for
for epoch ∈ [1, ..., T'] do
    for i∈ [1, ..., n] do
        v_i ← f_extract(x_i; θ)
    end for
    for i,j∈ [1, ..., n^2] do
        a_ij ← value of (10)
    end for
    W_0 ← A + A^T
    W ← D^{-1/2} W_0 D^{1/2}
    M ← value of (11)
    Z ← solve equation (19) by CG
    normalize Z
    generate pseudo-label for unlabeled data by (21)
    resample X_selected whose y'_i ≠ −1 and augment
    for i ∈ [1, n] do calculate ω_i by (22).
    for j ∈ [1, c] do calculate ξ_j by (|L_j| + |U_j|)^{-1}.
    θ ← OPTIMIZE(L_all)
end for
```

Our experiments are carried out on a workstation equipped with an Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz, two NVIDIA Geforce GTX 1080Ti GPUs with 11178MB VRAM and 236 GB RAM. The modules in the experiments are implemented by Python 3.6.12 with Pytorch 1.1.0.

4.1. DataSet

CIFAR-10 [38]. The dataset CIFAR-10 consists of 60,000 32 *x* 32 color images which are divided into 10 categories. Each category contains 6000 images. CIFAR-10 is divided into 50,000 training images and 10,000 test images. For each category, we respectively choose 10, 25, 50, and 100 labeled images, corresponding to 100, 250, 500, and 1k labeled data in the whole dataset, and the other images are regarded as unlabeled data. As all the categories contains the same number of images, we can select the same percentage of labeled data for each category based on this strategy.

CIFAR-100 [38]. The dataset CIFAR-100 also includes 60,000 32×32 color images, but those are divided into 100 categories. Each category contains 600 images. The division of the training and test dataset are the same as CIFAR-10. For each category, we choose 10, 25, 40 labeled images as labeled data, and the remaining images are regarded as unlabeled data.

miniImageNet [34]. The dataset miniImageNet includes 60,000 84×84 color images, which is a lightweight version of ImageNet. Those images are divided into 100 categories. Each category contains 600 images. The division of the training and test dataset are the same as CIFAR-10 and CIFAR-100. For each category, we choose 40, 100 labeled images as labeled data, and the remaining images are regarded as unlabeled data.

COVID19-Xray. The dataset COVID19-Xray consists of 18,765 256×256 Xray images which are divided into 4 categories. Specifically, it includes 616 COVID-19 positive cases along with 10,192 Normal, 6012 Lung Opacity (Non-COVID lung infection) and 1345 Viral Pneumonia images. The specific preprocessing

method of the dataset and the schema of dividing the train, validation and the test set will be described in detail in Section 4.5.

4.2. Experimental configuration

Network structure. Our network structure is shown in Fig. 1. On CIFAR-10 and CIFAR-100 we use 13-layer CNN structure as our feature extraction module, and add a dropout [14] layer after each convolution layer. At the same time, we add one or more MLPs [8] as the projection module. For the COVID19-Xray dataset, we choose the ResNet-18 structure as the feature extraction module. For miniImageNet, we use ResNet-50 to extract features. Different from MT_LP, our network adds the projection modules and the contrastive loss for the feature extraction module, and adopt a partial label propagation strategy in the second stage.

Optimizer. Based on extensive experiments, we use the RAdam [39] optimizer only in the case of 100 labeled data on the CIFAR-10, and use the SGD optimizer in other cases. In certain circumstances, the RAdam optimizer works better. This will be analyzed in the ablation study.

Hyper-parameters. For both the first and second stages of our algorithm, we set epochs to 240 in most cases. We set the initial learning rate of the first stage as follows: 0.05 for CIFAR-10, 0.2 for CIFAR-100, 0.15 for COVID19-Xray, 0.05 for miniImageNet. The learning rate decays as the epoch increases. With the aid of the simulated annealing algorithm, its epoch is 260. In addition, we monitor the accuracy on the test set to achieve double restriction on the learning rate. We set the initial learning rate of the second stage: 0.05 for CIFAR-10, COVID19-xray and miniImageNet, 0.001 for CIFAR-100, attenuation and monitoring methods are the same as the first stage. We set the batch-size: 100 for CIFAR-10, 128 for CIFAR-100, COVID19-Xray and miniImageNet. We set the number of labeled data for each batch: 50 for CIFAR-10 and, 31 for CIFAR-100, COVID19-Xray and miniImageNet. The second stage is the label propagation stage, we set $k = 50$ for constructing a kNN graph, and set μ in Eq. (12) to 0.99. The propagation threshold is related to the number of target categories and the average entropy of all data, we set it to 2.0 for CIFAR-10, 4.0 for CIFAR-100, 1.3 for COVID19-Xray, and 4.5 for miniImageNet.

Data augmentation. For weak data augmentation, we first randomly shift n pixels of the image vertically or horizontally through the transpose and crop functions of the Image class of PIL package. Here, n is set to 4. Secondly, we randomly flip the image horizontally through the RandomHorizontalFlip function of the transforms package. For strong data augmentation, we utilize the ColorJitter function of the transforms package to change the image pixels, where the three parameters brightness, contrast, and hue are all set to 0.5. The Strong&weak data augmentation is the combination of these two types of data augmentation.

4.3. Competing algorithms

As the proposed method CL_PLP is a transductive deep semi-supervised algorithm based on partial label propagation and self-supervised strategy, we compare our algorithm with two state-of-the-art transductive deep semi-supervised methods DLP [7] and MT_LP [7], six classical deep semi-supervised algorithms such as PL [17], Π-model [13], Temporal [13], VAT [15], MT [32] and ICT [40]. In addition, we also compare our algorithm with two classical supervised models Inception-V3 and ResNet18 and a fully supervised baseline with different amounts of available labeled data .

DLP utilizes the labeled data to pretrain a network in the first stage, and utilizes this pre-trained network to extract features of the whole data for further fully label propagation. The algorithm Mean Teacher (MT) uses two noised data as input, calculates the exponential moving average of weights at each training iteration, and compares the resulted final-layer activation using different network parameters. MT_LP is an algorithm that integrates MT and DLP. VAT approximates the perturbation to the corresponding input data that would yield the largest change in the network output, and then incorporates a loss term into the loss function to penalize the difference in the network outputs for perturbed and unperturbed input data. Π-model trains two perturbed neural network models by using dropouts in the perturbation process, and penalizes the differences in the final layer activation of the two networks. Temporal combines multiple perturbations of a network model, and compares the activation of the neural network at each epoch with the activation of the network at previous epochs. PL constructs a single supervised classifier that is iteratively trained on both labeled data and pseudo-labeled data labeled by the algorithm during the previous iterations. ICT constructs an interpolation of unlabeled data by mixup and encourages the prediction of unlabeled data to be consistent with the prediction of interpolation.

For the supervised models Inception-V3 and ResNet18, we utilize the classical supervised learning loss cross-entropy as the training loss, which can be seen from Eq. (2). For fully supervised baseline in Table 1, we implement it based on MT. Specifically, the implementation details are as follows: only use labeled data as the inputs of network, and retain the supervised and consistency loss of labeled data during training. Similarly, the data augmentation in our proposed method is also adopted in these supervised algorithms.

4.4. Performance comparison

To validate the effectiveness of our proposed method CL_PLP, we compare it with the above-mentioned state-of-the-art algorithms on two standard benchmark datasets CIFAR-10 and CIFAR-100. From Table 1, we observe that the accuracy of our algorithm is higher than the competing methods in most cases with different number of labeled data. Here, our implementation of ICT is different from the original one. The main difference is that we improve the unsupervised loss by adopting CE instead Consistence function for unlabeled data, so that, when we use 4000 labeled data on CIFAR-100, ICT slightly outperforms our algorithm. In the case of training with the least label, such as 100 labeled data for CIFAR-10 and 1000 labeled data for CIFAR-100, our method significantly improves over state-of-the-art. Compared with the second method MT_LP, the accuracy of our method is higher by about 25%, 11% respectively. For 250 labeled images (25 images per class) on CIFAR-10, the classification accuracy is 12% higher than that of MT_LP [7]. With the increase of labeled data known, the improvement in accuracy will be slowed down, as shown in Fig. 3. When using 500 and 1000 labeled images, the accuracy increases by about 5% and 3% respectively. When 4000 labeled images are used for CIFAR-100, the accuracy of ICT is slightly higher than that of our method by about 2%. In summary, when the number of labeled data is very small, the strength of our algorithm is relatively dominant.

As observed from Table 1, in the case of different numbers of labeled data, the supervised methods that only use labeled data are not competitive compared to the semi-supervised methods that use both labeled and unlabeled data. Compared with the supervised methods, our semi-supervised method obtains a significant improvement in accuracy (about 50% and 30%), especially when the available labeled data is rare (100 for CIFAR-10 and 100 for CIFAR-100). These results further indicate the importance of exploiting the information of the unlabeled data to improve the classification performance.

We also study the performance on miniImageNet. From Table 2, we compare the error rate with fully supervised baseline,

Table 1
Performance comparison between our algorithm and ICT, MT_LP, MT, DLP, PL, VAT, Temporal, Π-model and fully supervised baseline method. The performance is evaluated by the error rate, and calculated as the average of five experiments.

Dataset	CIFAR-10				CIFAR-100		
Labeled data per category	10	25	50	100	10	25	40
Inception-V3 [41]	78.44 ± 0.12	69.95 ± 0.34	63.28 ± 0.02	52.72 ± 0.13	87.26 ± 0.06	71.28 ± 0.08	63.14 ± 0.16
ResNet18 [42]	76.34 ± 0.06	66.61 ± 0.05	59.38 ± 0.04	53.56 ± 0.04	87.10 ± 0.05	78.45 ± 0.07	73.61 ± 0.07
Full supervised	71.57 ± 0.32	57.08 ± 0.10	35.17 ± 2.46	23.79 ± 1.31	79.90 ± 0.07	67.61 ± 0.18	57.93 ± 0.25
Π-model [13]	60.29 ± 1.14	46.48 ± 1.29	38.31 ± 1.27	25.52 ± 0.72	77.99 ± 1.33	60.56 ± 0.73	51.09 ± 0.16
Temporal [13]	54.86 ± 2.23	43.25 ± 2.51	29.47 ± 0.65	20.36 ± 0.41	74.20 ± 0.20	59.58 ± 0.41	52.58 ± 0.15
VAT [15]	62.86 ± 3.12	50.52 ± 1.73	38.66 ± 1.26	26.19 ± 0.48	73.80 ± 0.37	59.90 ± 0.65	51.07 ± 0.06
PL [17]	57.63 ± 1.82	48.72 ± 1.54	34.07 ± 2.37	25.03 ± 0.94	75.78 ± 0.33	58.21 ± 1.97	51.63 ± 2.27
DLP [7]	66.41 ± 0.22	48.42 ± 0.91	32.40 ± 1.80	22.00 ± 0.88	79.40 ± 0.31	53.91 ± 0.05	46.20 ± 0.76
MT [32]	56.14 ± 0.70	47.32 ± 2.30	27.45 ± 2.64	19.04 ± 0.51	67.03 ± 0.14	53.91 ± 0.57	45.36 ± 0.49
MT_LP [7]	52.20 ± 0.57	34.30 ± 0.90	24.02 ± 2.44	16.93 ± 0.70	66.79 ± 0.12	51.78 ± 0.16	45.07 ± 0.10
ICT [40]	42.62 ± 1.42	28.67 ± 1.25	21.16 ± 0.93	14.83 ± 0.17	64.63 ± 0.84	51.28 ± 0.96	**41.76 ± 0.36**
CL_PLP	**27.97 ± 0.07**	**21.84 ± 0.17**	**19.97 ± 0.12**	**14.14 ± 0.09**	**55.89 ± 0.11**	**49.98 ± 0.09**	**43.44 ± 0.19**

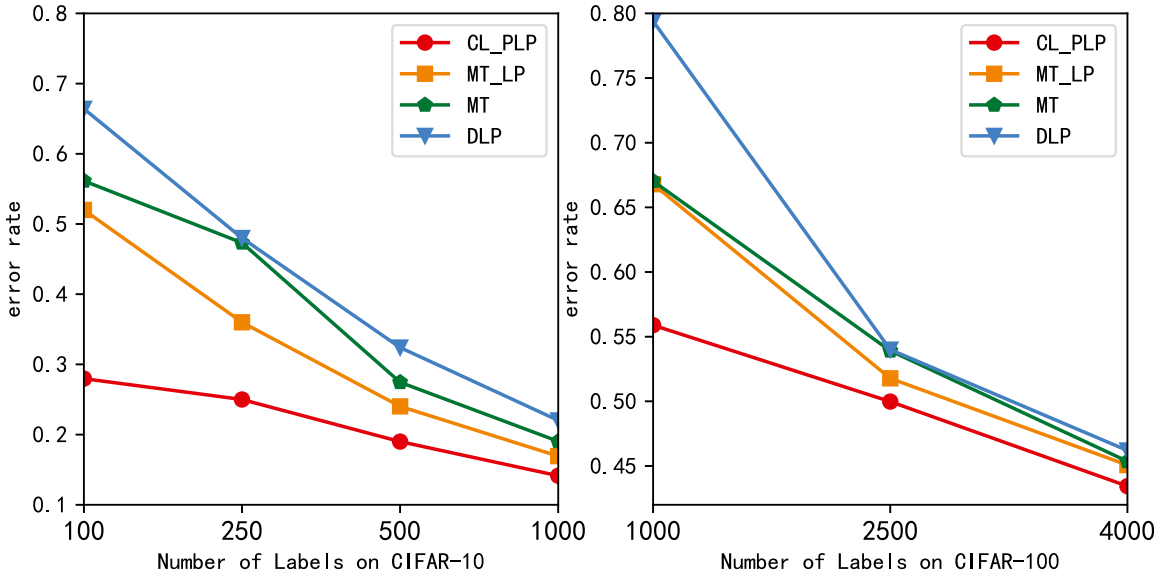

Fig. 3. Error rate versus number of labeled images on CIFAR-10 and CIFAR-100 using different methods.

Table 2
The error rate of Fully supervised, DLP,MT, MT_LP, ICT and our proposed method on miniImageNet.

Labeled data per category	40	100
Fully supervised	74.78 ± 0.33	60.25 ± 0.29
DLP [7]	70.29 ± 0.81	57.58 ± 1.47
MT [32]	72.51 ± 0.22	57.55 ± 1.11
MT_LP [7]	72.78 ± 0.15	57.35 ± 1.66
ICT [40]	66.64 ± 0.33	44.81 ± 0.76
CL_PLP	**50.84 ± 0.22**	**41.94 ± 0.53**

MT, DLP, MT_LP and ICT methods. For previous studies MT, DLP and MT_LP, we show the error rate that has been printed in these papers, and for ICT, Fully supervised and our proposed method, we take the average of the minimum of five experiments. Compared with the best algorithm ICT among competing algorithms, the accuracy of our proposed method improves about 16% in the case of 40 labeled data available each class, and also improves about 3% in the case of 100 labeled data available each class.

4.5. Application to COVID19-Xray dataset

Since COVID19 broke out in late 2019, COVID19-Xray image recognition and classification has become an urgent and challenging topic. The difficulty mainly lies in two aspects, including inconsistent sources and serious class-imbalance [43–45]. In our experiment, we retrieve the new dataset from Kaggle[1], which is a public website for COVID19-Xray datasets. The dataset integrates many datasets from different platforms and removes the duplicates. To tackle the class-imbalance problem, we select weak augmentation to over-sample the COVID19 images, and under-sample normal images. We regard those lung-Opacity and Viral Pneumonia images as other lung disease images (no COVID19 infection), to achieve data balance.

Subsequently, we firstly divide the dataset into training, validate and test set with the proportion 6:2:2. The training set is used to train the neural network model. The validation set is used to verify the validity of the trained model, and the model with the best effect is selected until we get a satisfactory model. Finally the test set is utilized to evaluate the effectiveness of the trained model. We select 10%, 20%, and 30% labeled data as different proportions of labeled data available for experiments. It is worth noting that we use stratified sampling to complete the above preprocessing. The purpose is to maintain the proportion of data with different class in the training set, validate set, test set and the labeled data available. As shown in Fig. 4, when some labeled data are selected, such as 30%, 20% and 10%, our semi-supervised classification algorithm outperforms that of

[1] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

Table 3
The results of ablation experiments on CIFAR-10 with 100 labels. Aug represents the adopted data augmentation method. The second column records the number of sub-networks. The pro&con column represents whether to add a projector layer and the construction of the contrastive loss, and the diffusion column records the propagation way including full propagation and partial propagation. The optimizer column denotes the optimizer used in different stages. The p_label_acc column records the accuracy of the pseudo-label obtained in the second stage, and the last column is the classification accuracy (top1) of the algorithm.

Aug	MT	Stage	Pro&con	Diffusion	Optimizer	p_label_acc	acc_top1
weak	FALSE	1	no	no	RAdam	no	24.54
Strong&weak	FALSE	1	no	no	RAdam	no	33.53
Strong&weak	FALSE	1	no	no	RAdam	no	**38.20**
weak	TRUE	1	no	no	RAdam	no	45.77
Strong&weak	TRUE	1	no	no	RAdam	no	54.11
Strong&weak	TRUE	1	yes	no	SGD	no	58.50
Strong&weak	TRUE	1	yes	no	RAdam	no	**61.60**
weak	TRUE	2	no	full	SGD	0.38	46.92
weak	TRUE	2	yes	partial	SGD	**0.58**	46.59
Strong&weak	TRUE	2	yes	full	SGD	0.63	65.77
Strong&weak	TRUE	2	yes	partial	RAdam	0.83	**68.50**
Strong&weak	TRUE	2	yes	partial	SGD	**0.88**	**72.60**

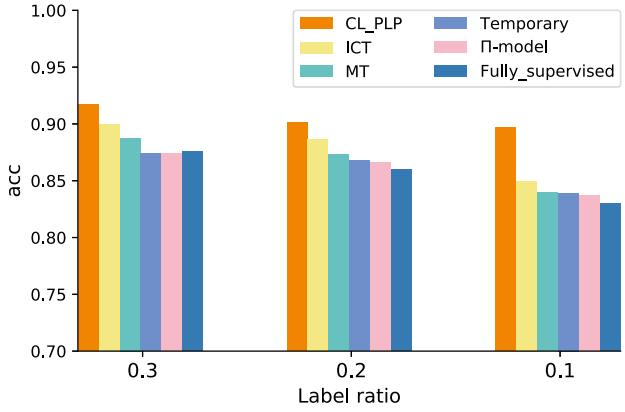

Fig. 4. Performance comparison between the fully-supervised, Π-model, Temporal, MT, ICT and our proposed method on the COVID19-Xray dataset. The horizontal axis represents the proportion of the labeled data used in the total data, and the vertical axis represents the accuracy.

full supervised learning in the same case. Especially when only 10% labeled data are used, the classification accuracy of semi-supervised algorithm can reach 89.7%, which is about 7% higher than that of full supervision algorithm, which shows the potential of our algorithm in other learning fields. Meanwhile, we compare the classification accuracy of our algorithm with those of the classical semi-supervised algorithms MT, ICT, Temporal, and Π-model on the COVID19-Xray dataset with different proportions of labeled data. The experimental results demonstrate our proposed method achieves better performance than other semi-supervised algorithms in all cases. Especially when 10% of the labeled data is available, our method improves the accuracy at least 5%–6% compared to other methods. The less labeled data is available, the more pronounced the advantage of the proposed approach is.

4.6. Ablation study

In order to investigate the impact of different components of our algorithm on the performance, we conduct two ablation studies, respectively under the conditions of 10 labeled images in each category of CIFAR-10 and 10, 40 and 100 labeled images in each category of CIFAR-100. In particular, the former experiment validates the effectiveness of Strong&weak data augmentation, projection layer and partial label propagation, while the latter experiment validates the sensitivity of CIFAR-100 to Strong&weak data augmentation.

Table 3 shows the performance comparison on CIFAR-10 with 100 labels. In the first stage, the accuracy of the model pre-trained by DLP+Strong&weak data augmentation increases by about 15% compared with DLP [7]. The accuracy of the MT_LP+Strong&weak data augmentation is about 9% higher than that of the MT_LP [7]. These results imply that the augmentation method which we use can avoid the overfitting problems caused by inadequate labels. Then, by adding the projector layer and constructing the contrastive loss, the accuracy of the algorithm increases by 7%–8%. It proves that Strong&weak, adding the projection layer and constructing the symmetrical contrastive loss can play a positive role in the training of the network (CIFAR-10 dataset). In the second stage, we focus on the ablation of partial and full propagation. When partial propagation is used directly in the MT_LP algorithm, although the pseudo-labels propagated increase by 20%, the accuracy decreases. This is because the pre-trained model whose accuracy is too low to obtain few high-quality pseudo-labels. When we use the improved pre-trained model, the accuracy of pseudo-labels increases by 20%–25% using partial propagation compared with full propagation. The classification accuracy rate (72.6%) is about 11% higher than the pre-trained model (61.6%), and about 25% higher than the original algorithm (46.92%). It can be seen that the accuracy of the algorithm is the highest when the combination of strong and weak augmentation, and the partial propagation strategy are used simultaneously. In addition, we also study ablation of optimizer used. It can be seen that in the first stage, under the same circumstances, using RAdam (61.60%) is better than SGD (58.5%). In the second stage, using SGD (72.60%) is better than RAdam (68.5%).

The second ablation is based on CIAFR-100. We mainly focus on the influence of the image augmentation method and the number of projector layers. Table 4 shows the classification performance comparison on CIFAR-100 test set. With different proportions of the labeled data, the augmentation method with strong and weak combination is not as effective as that with weak augmentation algorithm. For example, in the case of using 4000 labeled data and one projector layer, the accuracy of the weak augmentation method is 4% higher than that with strong and weak augmentation. As the size of CIFAR-100 images are 32×32, and there are a total of 100 categories, the difference in characteristics among different categories is very small. After strong augmentation, the pixel value will change, which has a negative impact on the following network training. On the contrary, we can observe that the projector layer on this dataset can indeed improve the accuracy of the algorithm. This observation is also different from CIFAR-10. Thus, for datasets with little difference in different categories, we recommend using weak augmentation and adding projector layers.

4.7. Integration study

Although our algorithm is a kind of transductive semi-supervised learning algorithm, it can be integrated with the

Table 4
The results of the ablation study on CIFAR-100. Here, three different numbers of labeled data from each category are selected for ablation, Aug represents the enhancement method, pro_num represents the number of added projection layers, and the performance is evaluated by classification accuracy (top1).

Labeled data per category	10				40				100			
Aug	weak	weak	weak	Strong&weak	weak	weak	weak	Strong&weak	weak	Strong&weak	weak	weak
pro_num	0	2	1	1	0	1	2	1	0	0	1	2
acc_top1	33.2	40.3	**44.5**	39.6	53.7	54.3	**55.2**	50.1	62.3	58.95	63.9	**64.5**

Table 5
The performance of our method integrated with MixMatch and FixMatch on CIFAR-10. The methods with the suffix "_PLP" are the methods after integration. The performance is evaluated by classification accuracy (top1).

Labeled data per category	10	25	50
MixMatch [5]	82.85	88.43	90.35
MixMatch_PLP	**85.57**	**89.53**	**90.89**
FixMatch [6]	93.49	94.22	95.34
FixMatch_PLP	**94.33**	**95.04**	**95.37**

existing inductive semi-supervised learning algorithm. Based on the proposed method, we further propose an improvement strategy for inductive semi-supervised learning. Specifically, we use an inductive semi-supervised algorithm as our pretraining stage to get a better network, and then conduct the partial label propagation process of our algorithm. By integrating our partial label propagation module, inductive semi-supervised learning methods can obtain high-quality pseudo-labels of unlabeled data and consequently improve their performance.

Our integration strategy is respectively applied to MixMatch [5] and FixMatch [6], which are two state-of-the-art inductive semi-supervised learning methods. The performance comparisons on CIFAR-10 are listed in Table 5. We conduct the integration experiments on the CIFAR-10 dataset using 100, 250, and 500 labeled data. As shown in Table 5, when the number of available labeled samples is very small, that is, only 10 labeled samples are used for each class, the performance of MixMatch and FixMatch is both improved by applying our integration strategy. The accuracy of MixMatch integrated with our partial label propagation module, that is MixMatch_PLP(85.57%), is around 3% higher than MixMatch (82.85%). Similarly, FixMatch integrated with the partial label propagation module (FixMatch_PLP : 94.49%) is also about 1% higher than the original algorithm (93.49%). When using 250 and 500 labeled data, the performance of MixMatch and FixMatch are also improved. Overall, the performance improvement is more obvious when the number of available labels is smaller, which proves the potential of our algorithm in integrating the existing inductive learning methods.

5. Discussion and future work

This paper proposes a semi-supervised transductive algorithm based on self-supervised contrastive learning and partial label propagation strategy. The proposed method consists of two modules, including feature extraction module and partial label propagation module. We propose an improved network structure, adding a projection layer module and contrastive loss for contrastive learning. We expand the dataset by combining strong and weak augmentation to increase the diversity of the dataset and the robustness of the model. By interrupting the label propagation according to the quality of pseudo-labels, the impact of high-quality pseudo-labels are improved, which is helpful for the following training. Finally, we propose a process integrated with state-of-the-art inductive semi-supervised algorithms.

To validate the proposed method, we conduct extensive experiments on three standard baseline datasets and a medical dataset COVID19-Xray. The performance of our algorithm on the classical datasets CIFAR-10, CIFAR-100 and miniImageNet with very few labeled data (e.g. 100 labels for CIFAR-10, 1000 labels for CIFAR-100, 4000 labels for miniImageNet) is significantly better than state-of-the-art transductive deep semi-supervised algorithms. We also apply our algorithm to the medical COVID19-Xray dataset, it still shows good performance. When using 30% labeled data, the accuracy is about 91%. We also conduct ablation studies to verify the impact of different parts of our algorithm on performance.

Finally, in view of the recent rapid development of inductive deep semi-supervised learning, we propose a strategy to integrate our method with inductive learning methods, and the results demonstrate that it can further improve the accuracy and obtain additional high-quality pseudo-labels of the unlabeled dataset for training the deep neural networks. In the future, we plan to focus on exploring more effective and efficient self-supervised ways to extract features from labeled and unlabeled data, and transferring our algorithm to real application scenarios.

CRediT authorship contribution statement

Yanglan Gan: Investigation, Conceptualization, Methodology, Writing – review & editing. **Huichun Zhu:** Methodology, Software, Data curation, Formal analysis, Validation, Writing – original draft. **Wenjing Guo:** Supervision, Validation, Resources, Funding acquisition. **Guangwei Xu:** Writing – review & editing. **Guobing Zou:** Visualization, Writing – reviewing and editing.

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 was supported by National Natural Science Foundation of China (No. 62172088, 61772128), Shanghai Natural Science Foundation (No. 21ZR1400400, 19ZR1402000), and National Key Research and Development Program of China (2017YFC0907505).

References

[1] Longlong Jing, Yingli Tian, Self-supervised visual feature learning with deep neural networks: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (2020).
[2] Jesper E. van Engelen, Holger H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2) (2020).
[3] Xiaojin Zhu, John Lafferty, Zoubin Ghahramani, Semi-Supervised Learning: From Gaussian Fields to Gaussian Processes, School of Computer Science, Carnegie Mellon University, 2003.
[4] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, Bernhard Schölkopf, Learning with local and global consistency, in: Advances in Neural Information Processing Systems, 2004, pp. 321–328.
[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, Colin A Raffel, MixMatch: A holistic approach to semi-supervised learning, Adv. Neural Inf. Process. Syst. 32 (2019).
[6] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, Colin Raffel, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, Adv. Neural Inf. Process. Syst. (2020).

[7] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Ondrej Chum, Label propagation for deep semi-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5070–5079.
[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.
[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, Geoffrey E Hinton, Big self-supervised models are strong semi-supervised learners, Adv. Neural Inf. Process. Syst. 33 (2020) 22243–22255.
[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, Ross Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.
[11] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, Lucas Beyer, S4l: Self-supervised semi-supervised learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1476–1485.
[12] Enislay Ramentol, Yailé Caballero, Rafael Bello, Francisco Herrera, SMOTE-RS B*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory, Knowl. Inf. Syst. 33 (2) (2012) 245–265.
[13] Laine Samuli, Aila Timo, Temporal ensembling for semi-supervised learning, in: International Conference on Learning Representations, Vol. 4, ICLR, 2017, p. 6.
[14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[15] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Shin Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2018) 1979–1993.
[16] Abhinav Shrivastava, Saurabh Singh, Abhinav Gupta, Constrained semi-supervised learning using attributes and comparative attributes, in: European Conference on Computer Vision, Springer, 2012, pp. 369–383.
[17] Dong-Hyun Lee, et al., Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: Workshop on Challenges in Representation Learning, Vol. 3, ICML, 2013, p. 896.
[18] Yves Grandvalet, Yoshua Bengio, et al., Semi-supervised learning by entropy minimization, CAP 367 (2005) 281–296.
[19] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, Nanning Zheng, Transductive semi-supervised deep learning using min-max features, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 299–315.
[20] Zahra Donyavi, Shahrokh Asadi, Diverse training dataset generation based on a multi-objective optimization for semi-supervised classification, Pattern Recognit. 108 (2020) 107543.
[21] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, W Philip Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.
[22] Alberto Fernández, Salvador Garcia, Francisco Herrera, Nitesh V Chawla, SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary, J. Artificial Intelligence Res. 61 (2018) 863–905.
[23] Spyros Gidaris, Nikos Komodakis, Dynamic few-shot visual learning without forgetting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4367–4375.
[24] Xinlei Chen, Haoqi Fan, Ross Girshick, Kaiming He, Improved baselines with momentum contrastive learning, 2020, arXiv preprint arXiv:2003.04297.
[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al., Bootstrap your own latent: A new approach to self-supervised learning, 2020, arXiv preprint arXiv:2006.07733.
[26] Xinlei Chen, Kaiming He, Exploring simple siamese representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15750–15758.
[27] Guocheng Wang, Zhengyou Xia, Label propagation algorithm based on topological potential, in: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Springer, 2020, pp. 131–143.
[28] Jonatan Møller Nuutinen Gøttcke, Arthur Zimek, Ricardo JGB Campello, Non-parametric semi-supervised learning by Bayesian label distribution propagation, in: International Conference on Similarity Search and Applications, Springer, 2021, pp. 118–132.
[29] Bo Li, Qiang Zheng, Kun Zhao, Honglun Li, Chaoqing Ma, Shuanhu Wu, Xiangrong Tong, Multi-atlas segmentation combining multi-task local label learning and semi-supervised label propagation, in: International Conference on Image and Graphics, Springer, 2021, pp. 762–772.
[30] Liang Bai, Junbin Wang, Jiye Liang, Hangyuan Du, New label propagation algorithm with pairwise constraints, Pattern Recognit. 106 (2020) 107411.
[31] Ismail Elezi, Alessandro Torcinovich, Sebastiano Vascon, Marcello Pelillo, Transductive label augmentation for improved deep network learning, in: 2018 24th International Conference on Pattern Recognition, ICPR, IEEE, 2018, pp. 1432–1437.
[32] Antti Tarvainen, Harri Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, Adv. Neural Inf. Process. Syst. 30 (2017).
[33] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, Alan Yuille, Deep co-training for semi-supervised image recognition, in: Proceedings of the European Conference on Computer Vision, Eccv, 2018, pp. 135–152.
[34] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al., Matching networks for one shot learning, Adv. Neural Inf. Process. Syst. 29 (2016) 3630–3638.
[35] Mehdi Sajjadi, Mehran Javanmardi, Tolga Tasdizen, Mutual exclusivity loss for semi-supervised deep learning, in: 2016 IEEE International Conference on Image Processing, ICIP, IEEE, 2016, pp. 1908–1912.
[36] Fan R.K. Chung, Fan Chung Graham, Spectral Graph Theory, American Mathematical Soc, 1997.
[37] Jonathan Richard Shewchuk, et al., An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, Carnegie-Mellon University. Department of Computer Science, 1994.
[38] Alex Krizhevsky, Geoffrey Hinton, et al., Learning Multiple Layers of Features from Tiny Images, Citeseer, 2009.
[39] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, Jiawei Han, On the variance of the adaptive learning rate and beyond, 2019, arXiv preprint arXiv:1908.03265.
[40] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, David Lopez-Paz, Interpolation consistency training for semi-supervised learning, Neural Netw. (2021).
[41] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[43] Ezz El-Din Hemdan, Marwa A. Shouman, Mohamed Esmail Karar, Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images, 2020, arXiv preprint arXiv:2003.11055.
[44] Michael Roberts, Derek Driggs, Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, et al., Machine learning for COVID-19 detection and prognostication using chest radiographs and CT scans: a systematic methodological review, 2020, arXiv preprint arXiv:2008.06388.
[45] Angelica I Aviles-Rivero, Philip Sellars, Carola-Bibiane Schönlieb, Nicolas Papadakis, GraphXCOVID: explainable deep graph diffusion pseudo-labelling for identifying COVID-19 on chest X-rays, Pattern Recognit. 122 (2022) 108274.