Robust semi-supervised classification based on data augmented online ELMs with deep features

Xiaochang Hu[a], Yujun Zeng[a], Xin Xu[a,b,*], Sihang Zhou[a], Li Liu[c,d]

[a]*College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China*
[b]*Laboratory of Science and Technology on Integrated Logistics support, National University of Defense Technology, Changsha 410073, China*
[c]*College of System Engineering, National University of Defense Technology, Changsha 410073, China*
[d]*Center for Machine Vision and Signal Analysis, University of Oulu, Finland*

Abstract

One important strategy in semi-supervised learning is to utilize the predicted pseudo labels of unlabeled data to relieve the overdependence on the ground truth of supervised learning algorithms. However, the performance of such kinds of semi-supervised methods heavily relies on the quality of pseudo labels. To address this issue, a robust semi-supervised classification method, named data augmented online extreme learning machines (ELMs) with deep features (DF-DAELM) is proposed. This method firstly extracts feature representation and infers labels for unlabeled data through self-training. Then, with the learned features and inferred labels, two noise-robust shallow classifiers based on data augmentation (i.e., SLI-OELM and CR-OELM) are proposed to eliminate the adverse effects of noises on classifier training. Specifically, inspired by label smoothing, a data augmented method, SLI-OELM is designed based on stochastic linear interpolation to improve the robustness of classifiers based on ELMs. Furthermore, based on the smoothing assumption, the proposed CR-OELM utilizes an ℓ_2-norm consistency regularization term to implicitly weight noisy samples. Comprehensive experiments demonstrate that DF-DAELM achieves competitive or even better performance on CIFAR-10/100 and SVHN over the

*Corresponding author
Email address: `xinxu@nudt.edu.cn` (Xin Xu)

related state-of-the-art methods. Meanwhile, for the proposed classifiers, experimental results on the MNIST dataset with different noise levels and sample scales demonstrate their superior performance, especially when the sample scale is small ($\leq 20K$) and the noise is strong ($40\% \sim 80\%$).

Keywords: deep semi-supervised learning, extreme learning machine, noise-tolerant, data augmentation

---

1. Introduction

In the past decades, with the improvement of network designing techniques, the leaping of computational power, and the accumulation of large-scale high-quality labeled data, deep learning has achieved remarkable performance in many machine learning applications and attracted the attention of many researchers in various fields [1, 2, 3]. However, as a general artificial intelligence method, the overdependence on a large amount of high-quality labeled data limits these algorithms from having a larger impact in more fields. As a consequence, training the networks better with less human guidance is becoming a hot research spot in the field of deep learning, and semi-supervised learning (SSL) is one of the important directions. Deep SSL requires achieving preferable performance with a small number of labeled data and unlimited easily available unlabeled data. Many researches have been done in this direction and the existing popular algorithms can be roughly categorized into two categories. The first category is pseudo-label-based methods, which estimate the pseudo labels of the unlabeled data and adopt them as extra supervision to exploit discriminative information from the whole dataset [4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. The second category is pre-training-based methods. These methods pre-train the deep neural network to find compressed representations of input data with auxiliary unsupervised tasks before training the classifier with labels [2, 14, 15, 16, 17].

One of the representative methods in the first category is pseudo-labeling (self-training) [4, 18, 5], which reduces the overlap of the class probability distribution of both labeled and unlabeled data by minimizing the distance between true labels and pseudo labels. Another representative method is consistency regularization, which extracts the abstract invariance within the unlabeled data relying on the smoothing assumption that small perturbations for each sample should not significantly change the predicttion [9, 10]. However, these methods heavily rely on the quality of the predicted pseudo labels and will easily suffer from confirmation bias where the prediction errors would accumulate [18, 10, 19, 19, 7]. Recently, despite the fact that a variety of methods have been proposed to solve this problem, such as MeanTeacher [10] based on model ensembling, VAT [8] based on data perturbation, [20] based on meta learning and so on [7, 21, 12, 22, 13, 13, 23, 24, 19], confirmation bias is still an intractable issue in the field of deep SSL.

The second category generally consists of two stages, i.e., network pre-training and classifier learning [25]. Normally, the first stage finds the deep feature representation of the input data with excellent generalization performance through unsupervised or self-supervised algorithms [26, 27, 2, 28]. In the second stage, it usually adopts supervised fine-tuning [25] or traditional semi-supervised classifiers [14, 17, 29, 30, 16] to further enhance the discriminative capability of the learned feature and learn the final classifier. Since network pre-training is task-agnostic, the representations generated by the network pre-training are likely to be suboptimal for the ultimate classification tasks [31, 19, 32, 33]. Nevertheless, the pre-training-based methods are less sensitive to the confirmation bias thanks to its decoupling learning scheme, which pursues the outstanding performance of each stage separately without considering the quality of pseudo labels. Various applications, such as traffic sign classification [34], long-tailed recognition [35] and so on [36, 17, 14, 37], have demonstrated the effectiveness of such decoupling learning scheme.

Inspired by the decoupling scheme of the second category of semi-supervised methods, our key insight is to sovle confirmation bias encountered by the pseudo-label-based semi-supervised methods via decoupling feature representation and classifier. However, since the learned features and inferred labels for unlabeled data through such semi-supervised training (the pseudo-label-based methods) generally contain some noise (as shown in Fig.2), it is difficult to significantly improve the performance by retraining common classifiers.

As a representative single layer feedforward neural networks algorithm, extreme learning machine (ELM) [38, 39, 40] is characterized by its high learning efficiency and generalization performance, which has been successfully applied to a wide range of domains, such as traffic sign classification [34], fingerprint recognition [41], hyperspectral image classification [42] and so on [30, 36, 37, 43, 44, 17, 14]. However, due to the unboundedness of the mean square error (MSE) criterion used in traditional ELMs, the performance of ELMs is extremely susceptible to noisy data [45]. In order to solve this problem, in recent years, many researchers design complex regularizations [39, 46, 47] to prevent ELMs from overfitting noisy samples, while many works develop various weighted or non-convex loss functions [48, 46, 45, 49, 50] for ELMs to punish the noisy samples. Most of the above methods assume that the noisy data obeys non-gaussian distribution and improve the robustness of ELMs by designing various techniques based on empirical studies. However, when both the learned features and the inferred labels are interfered by noise, the corresponding distribution is difficult to estimate. It may be unfriendly to directly migrate these methods to our problem. Fortunately, data augmentation [51], which generates new data from the vicinity of the original data to expand the dataset based on the Vicinal Risk Minimization principle [52], should be a promising choice.. Since data augmentation is task-independent, it is very convenient to combine with any method without considering data distributions. Inspired by this, we try to use data augmentation [48, 46, 47, 49] to improve the robustness of ELMs. As far as we know, there is almost no study on improving the robustness of ELMs from the perspective of data augmentation.

In this paper, we propose a robust semi-supervised classification method to solve confirmation bias [18, 10, 19], named data augmented online extreme learning machines with deep features (DF-DAELM). This method first decouples the self-training scheme to extracts task-oriented deep features as well as infers pseudo labels of unlabeled data. Then, in order to eliminate the impact of noise in these features and labels on the performance of ELMs, we apply data augmentation to ELMs and then propose two robust shallow classifiers from two different perspective of data augmentation [51, 52] (i.e., stochastic linear interpolation online extreme learning machine (SLI-OELM) and consistency regularization online extreme learning machine (CR-OELM)). Concretely, inspired by label smoothing [21], we come up with a data augmented method called SLI-OELM. It first conducts stochastic linear interpolation to augment the data and then uses them to train the ELM classifiers, which significantly strengthens the robustness of ELM classifiers. Furthermore, motivated by the smoothness assumption [25, 9, 10, 11], CR-OELM develops a consistency regularization term to constrain the parameter space of the ELM classifier, which is described as the ℓ_2-norm of the model's prediction distance between the original sample and the augmented sample in its neighborhood. Extensive experiments demonstrate that DF-DAELM achieves competitive or even better classification performance on 3 datasets (CIFAR-10/100 and SVHN) over the state-of-the-art methods. Meanwhile, for the SLI-OELM and CR-OELM, experiments demonstrate substantial improvements over 3 robust ELM methods on MNIST with different label noise levels and data scales. It is worth noting that SLI-OELM and CR-OELM have strong robustness in high label noise levels ($40\% \sim 80\%$) and small data scale ($\leq 20K$). The contributions of this work are summarized as follows:

- A novel deep semi-supervised classification method named DF-DAELM is presented. Different from the previous methods addressing confirmation bias, it decouples self-training scheme to extract features and infer pseudo labels combined with the proposed noise-robust ELM classifier to improve the performance. Comprehensive experiments demonstrate that DF-DAELM achieves competitive or even better performance over state-of-the-art deep SSL algorithms.

- Two new robust ELM classifiers (i.e., SLI-OELM and CR-OELM) based on data augmentation are proposed. To our knowledge, this is the first time to utilize data augmentation to enhance the noise robustness of ELM-s. Compared with the current robust extreme learning machines, they are robust on the training datasets with high label noise level ($40\% \sim 80\%$) and small sample scale ($\leq 2K$).

This paper is organized as follows. Section 2 shows some notations and related work. Section 3 firstly describes the overall framework of the proposed DF-DAELM, then followed by task-oriented feature representation 3.1 and pseudo label generation (3.1) as well as two data augmented ELM classifiers (SLI-OELM and CR-OELM)(3.2). Then, Section 4 presents the comprehensive experiments and analysises. Finally, Section 5 concludes this paper.

2. Notations and Related work

In this section, we first briefly introduce some important notations and then review the related work, including deep SSL and extreme learning machine (ELM).

2.1. Notations

Throughout this paper, for the deep SSL task, we are given a training dataset, $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$, where $\mathcal{D}_l$ is the labeled subdataset with l labeled instances $\{(x_1, y_1), \cdots, (x_l, y_l)\}$ and $\mathcal{D}_u$ is the unlabeled subdataset with $n-l$ instances $\{x_{l+1}, \cdots, x_n\}$. Usually, $n - l \geq l$, $x \in \mathbb{R}^D$ and $y \in \{0,1\}^{C\times 1}$ being the one-hot encoding ground-truth label corresponding to x, where D is the dimension of input space and C is the number of output class. Let $X = [x_1, \cdots, x_n]^T \in \mathbb{R}^{n\times D}$ be the data matrix and $Y = [y_1, \cdots, y_l]^T \in \{0,1\}^{l\times C}$ be the label matrix. For an arbitrary matrix $M \in \mathbb{R}^{n\times m}$, we denote its (i,j)-th entry, the j-th column of M by m_{ij}, m_j respectively. The squared Frobenius norm of M is $\left\|M\right\|_F^2 = Tr(M^T M)$, where $Tr(\cdot)$ denotes the trace operator and the inverse of matrix M is denoted by M^{-1}. For a vector $v \in \mathbb{R}^m$, the ℓ_2-norm of vector v is $\sqrt{v^T v}$, where v^T is the transpose of v. $\boldsymbol{I}$ denotes an identity matrix and $\mathbf{1}$ is a column vector with all the elements as one. $\|\cdot\|$ is for norm.

2.2. Deep semi-supervised learning

This subsection reviews the deep SSL methods closely related to this research. More comprehensive introductions and reviews of existing SSL approaches could be found in [25, 53].

Pseudo-labeling [18] (self-training) methods treat the model predictions as the pseudo labels for unlabeled samples, which are used in training with the cross-entropy. The methods on the basis of consistency regularization [25, 54] relies on the smoothing assumption that a classifier should output similar predictions for an unlabeled sample even after it is augmented, such as Π-Model [9]. However, the methods heavily rely on the quality of the pseudo labels and are therefore quite apt to suffer from from the confirmation bias [18, 10, 19], where the incorrect pseudo labels would accumulate and harm the model training. To solve this problem, various methods have been proposed. One way is to improve the reliability of the predicted pseudo labels. LP [5] utilizes the graph-based label propagation to enhance the reliability of pseudo labels. Temporal Ensembling and MeanTeacher [10, 11] take one of the two predictions as the target and uses exponential moving average of the historical predictions or model parameters for each unlabeled example to enhance the stability of the target prediction. On the other hand, many researches [8, 7, 21, 12] find that stochastic perturbations applied to unlabeled data may be inefficient in feature representation and use various advanced data augmentations for consistency regularization to improves representation capability, such as VAT [8], WCP [7], and mixup [21, 12], etc. Recently, many researches proposed a series of holistic approaches utilizing the dominant methods in SSL to improve the performance of semi-supervised model [22, 13, 20], such as MixMatch [13], ReMixMatch [23], fixMatch [24] and CoMatch [19].

Another popular class of SSL methods [25] is the pre-training-based method, which decouples feature representation learning and classifier learning. For the representation learning, auxiliary unsupervised tasks mainly use reconstruction loss (e.g., autoencoder [26]) or self-supervised contrastive learning (e.g., SimCLR [2], MoCo [28]) to improve the generalization capability of the deep features.

For the classifier optimization, it mainly adopts supervised fine-tuning [25] and traditional semi-supervised classifiers, such as semi-supervised support vector machine and semi-supervised extreme learning machine [14, 17, 29, 30, 16]. However, as no labeled guidance is introduced in the auxiliary feature pre-training tasks, the feature learning process of such methods is task-agnostic, so that usually learns weak discriminative features, resulting in a sub-optimal model [33, 31].

In this paper, the proposed DF-DAELM method is motivated by the above research work and it mainly differs from the existing related work in the following two aspects. Firstly, we do not resort to complex training skills to relieve the model's overfitting of noisy pseudo labels but decouple the feature representation and classifier training to improve generalization performance. Secondly, we introduce label information into the feature representation training process rather than use unsupervised learning methods to enhance the correlation between the feature representation and the ultimate task, thereby reducing the risk of a suboptimal model.

2.3. Extreme learning machine

Extreme learning machine (ELM) is an effective learning framework using single-layer feedforward neural networks proposed by Huang [18, 10, 19], which can be used as a classifier. Because of the limitation of space, the traditional ELMs (basic ELM [38, 39] and Online sequential-ELM [55]) related to this paper are placed in the appendix A. Since this article focuses on the robust ELMs, we briefly review them as follows.

To improve the robustness of ELMs under noisy label or noisy data/features, the common strategy is re-weighting loss function under different samples. For example, [39] proposed a regularized ELM with a two-stage weighted least square to enhance the robustness. Due to the lack of flexibility of fixed weights, more attention have been paid to design special loss functions. Horata et al. [56] used iteratively reweighted least squares (IRLS) algorithm to solve the Huber loss function without a regularization term. [48] used ℓ_1-norm constraint on loss function to solve model degradation caused by different distribution samples. [46] imposed structured sparsity penalty of the ℓ_{21}-norm to improve the robustness of ELM. Based on the application of orthogonal constraints in subspace, the weight orthogonalization of the output matrix [47] is used to improve the robustness of the ELM model. In recent years, the non-convex loss functions have become more and more attractive. Correntropy-based ELM [49] used nonlinear similarity to avoid the negative impact of noisy labels, while [50] applied non-convex loss function to give constant penalties to noisy labels to suppress their negative influence. [45] adopted a non-convex fraction loss function based on Laplacian kernel to improve robustness. The main drawback of these methods is that the loss functions are too complex to be optimized and such methods usually rely on empirical studies.

Unlike the aforementioned work, in this paper, we attempt to use the augmented data to promote the robustness of ELM classifiers. In this way, there is no need to laboriously design complex objective functions or regularizations, since data augmentation is usually task-agnostic. We have proposed two data augmented classifiers (SLI-OELM and CR-OELM). For SLI-OELM, it exploits stochastic linear interpolation to augment the data and smooth the noisy labels to improve the robustness of the ELM classifier. For CR-OELM, it utilizes a consistency regularization term to effectively evaluates the prediction difference between the original sample and the augmented sample in its neighborhood, implicitly detecting and punishing the sample with the noisy label. In addition, since data augmentation has been widely used in training deep neural networks, the two proposed classifiers are very convenient to collaborate with deep convolution features to solve the confirmation bias encountered by the pseudo-label-based SSL methods.

3. The Proposed Approach

The popular pseudo-label-based semi-supervised methods usually suffer from confirmation bias [18, 10, 19], where the incorrect predictions would be reinforced. Aiming at this problem, this paper proposes a robust semi-supervised classification approach, DF-DAELM. It firstly extracts feature representation and infers labels for unlabeled data through self-training. Then, with the learned features and inferred labels, two noise-robust shallow classifiers based on data augmentation (i.e., SLI-OELM and CR-OELM) are proposed to eliminate the adverse effects of noises on classifier training.

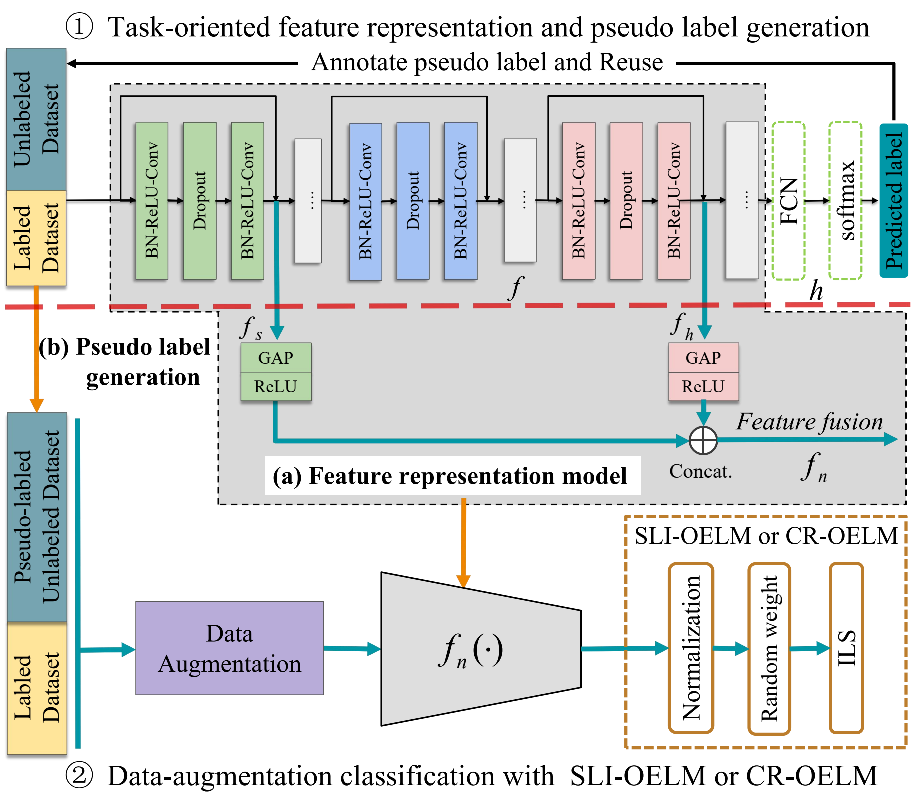


Figure 1: The overall framework of DF-DAELM. DF-DAELM consist of consists of two stages. For ① stage, it decouples a deep neural network (taking ResNet-18 [57] as an example here) by a self-training scheme (above the red dashed line) to obtain a task-oriented (a) feature representation model f_n that fusing the semantic feature f_h and shallow feature f_s, as well as (b) generate pseudo labels of unlabeled data (dark green rectangle) (Section 3.1). For ② stage, we take the features f_n of samples and pseudo labels as the input and target of the proposed robust ELMs based on data augmentation (SLI-OELM (Section 3.2.1) or CR-OELM (Section 3.2.2)) to improve the classification performance via retraining the classifiers (green solid line).

The pipeline of DF-DAELM is shown in Fig.1, including two stage: one is the pre-training phase stage mainly composed of task-oriented feature representation (Section 3.1) and pseudo-label generation (Section 3.1), and the other is design and retaining of noise-robust ELM classifiers (SLI-OELM (Section 3.2.1) and CR-OELM (Section 3.2.2)).

3.1. Task-oriented feature representation and pseudo label generation

First of all, this section introduces the two components (i.e., task-oriented feature extraction and pseudo label generation) of the pre-training phase stage of DF-DAELM.

Task-oriented feature representation. To avoid the degradation caused by the noisy pseudo labels, a straightforward idea is to discard the classifier and use unsupervised methods. However, it may learn representations that are suboptimal for the specific classification task, due to the task-agnostic unsupervised feature preprocessing [33, 51, 31]. In order to improve the discriminative capability and task consistency of the deep features, we propose to use the self-training method [4, 18] instead of unsupervised pre-training methods to pre-train the deep neural network. Concretely, we unify multiple regularizations (i.e., entropy regularization [18] and uniform distribution regularization [58]) to enhance the feature representation of self-training [59, 4, 18]. It is worth noting that the feature representation model here can be replaced with any other deep semi-supervised learning methods that encounter confirmation bias.

Formally, the deep feature representation $f(\cdot)$ is followed by a classification head (multilayer perceptron) $h(\cdot)$. The probability of the predicted label for the input can be denoted as follow.

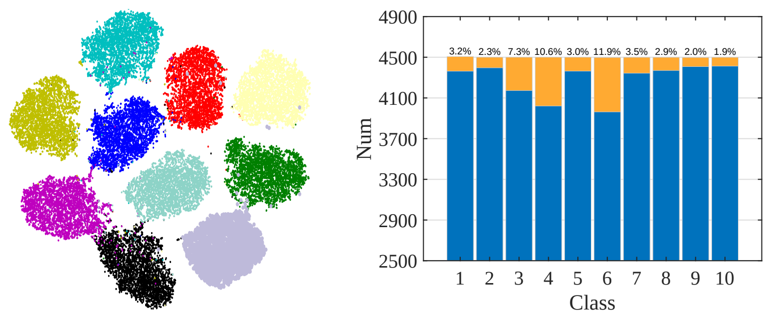
$$p\left(y|x\right) = softmax(h \circ f(x)) \tag{1}$$

The parameters of both $f(\cdot)$ and $h(\cdot)$ are iteratively optimized by minimizing the following loss function

$$l = -\sum_{i=1}^{l} y_i log\, p\big(y_i|x_i\big) - \lambda_0 \sum_{i=l+1}^{n} \tilde{y}_i log\, p\big(y_i|x_i\big) + \lambda_1 R_0 + \lambda_2 R_1 \tag{2}$$

where the first item is the loss of labeled samples, the latter is the loss of samples with pseudo labels, $\tilde{y}_i$ is the pseudo labels by hard assignment according to the prediction of the model $h \circ f(x_i)$. According to [58], we added two regularization items to improve the stability of network training. The first regularization term is $R_0 = \sum_{j=1}^{C} p_c\, log\big(\frac{p_c}{\hat{p}_j}\big)$, where $p_c = \frac{1}{C}$ is a uniform distribution and $\hat{p}_j$ denotes the mean $p\big(y_j|x\big)$ of the model for j-th class across all samples in the dataset. And then, in order to prevent the model from the local optimum, entropy regularization $R_1 = H(p(y|x))$ [18] is introduced. λ_0, λ_1 and λ_2 respectively represent the weighted coefficients of the loss of samples with pseudo labels and the two regularization terms.

The above is the feature representation learning process. However, although multiple regularizations are introduced, the feature representation $f(\cdot)$ may have a certain amount of noise, due to confirmation bias. As shown in Fig.2(a), the high-level semantic features (output by the last layer of $f(\cdot)$) of few samples are inseparable. It is not appropriate to directly use such features as the input of classifiers. Here, we give two solutions, one is feature fusion, the other is a noise-tolerant classifier. The second is our focus and will be introduced in detail in Section 3.2. As for the feature fusion, it is an alternative plan. Generally, the shallow features are not susceptible to the noisy labels [60, 61]. Thence we suggest fusing the shallow features and the high-level semantic features to relieve the feature-noise. As shown in Fig.1, for any sample x, its fusion feature is $f_n(x) = concat(ReLU(GAP(f_s(x))), ReLU(GAP(f_h(x))))$, where $f_s(\cdot)$ and $f_h(\cdot)$ represent the shallow features and the high-level semantic features respectively, GAP is global average pooling [62]. The ablation study of Section 4.2 demonstrated the effectiveness of feature fusion, indicating that it is a feasible solution to noisy feature .



(a) (b)

Figure 2: Statistics of features f_s and pseudo labels for training dataset of CIFAR-10 with 4000 labels (Section 3.1). (a): Feature visualization of the last layer of CNN with t-SNE and different colors represent the true label of each sample. (b): Statistics of the number of samples in each category for pseudo labels (Blue and yellow indicate correct and incorrect samples respectively.).

Pseudo label generation. For the second stage of the retraining of the classifier, it generally uses supervised fine-tuning or traditional SSL methods [29, 30]. However, they often suffer from poor classification performance (see the comparative experiment in section 4.2) and high solution costs [14, 17, 30, 17, 14, 16]. So, we propose to directly adopt the pseudo labels $\tilde{y}$ predicted and hard assigned by the final model $h \circ f(x)$ for the unlabeled samples, and convert the classifier optimization to a fully-supervised one to alleviate these problems. The label matrix of all samples is reformulated as $Y = [y_1, \ldots, y_l, \tilde{y}_{l+1}, \ldots, \tilde{y}_n]^T$, where $\tilde{y}$ is the predicted pseudo label for the unlabeled sample. Finally, after total samples $\mathcal{D}$ are processed, the input and target of the classifier in the second stage are $f_n(X)$ and Y respectively.

This operation is efficient and convenient, but unfortunately, there will be a small number of noisy features and labels, which brings challenges to the performance of the common to solve this problem.

3.2. SLI-OELM and CR-OELM classification with data augmentation

In this section, we propose two robust ELM classifiers. Compared with other common classifiers (such as SVM), ELM has the advantage of mitigating the noisy feature faced by our method due to the single hidden layer neural network. However, due to its nature of the squared loss function, incorrect labels will cause huge penalties and affect the stability of the decision hyperplane, resulting in performance degradation ELM classifiers [50]. With the development of deep learning, it is found that the data itself contains a variety of knowledge that is beneficial to improve the generalization of the model [52], such as data augmentation [51] plays an important role in enhancing the generalization. Inspired by this, we use data augmentation to directly explore the knowledge that exists in the data instead of loss function design [48, 46, 47, 49] to improve the robustness of ELMs. Specifically, we propose two robust ELM algorithms for DF-DAELM, namely the stochastic linear interpolation online extreme learning machine (SLI-OELM) and the consistency regularization online extreme learning machine (CR-OELM).

Note that in this section, $f_n(\cdot)$ is the trained feature representation model from the first stage (see Section 3.1) and $g(\cdot)$ is the output function of the random hidden layer after activation of ELM (Eq.(A.1)), used to replace the classification head $h(\cdot)$ of the first stage. And $g\big(f_n(\,\cdot\,)\big)$ means the composite function.

3.2.1. Stochastic linear interpolation online ELM (SLI-OELM)

In order to alleviate the performance degradation of the classifier caused by incorrect labels and improve generalization, we propose a new algorithm, that is the stochastic linear interpolation ELM (SLI-ELM), which uses stochastic linear interpolation to smooth the labels to prevent samples with the noisy label from disturbing the decision hyperplane. Concretely, we adopt the stochastic linear interpolation based on mixup data augmentation [21], which implicitly remedies the huge penalty resulting from noisy labels on the classifier by convex optimization that noisy labels and noise-free labels.

As shown in Eq.(3), we construct the convex combinations of sample pairs and corresponding labels.

$$\begin{aligned}\tilde{X} =&\Lambda X_i + (\boldsymbol{I} - \Lambda)X_j \\ \tilde{Y} =&\Lambda Y_i + (\boldsymbol{I} - \Lambda)Y_j\end{aligned} \tag{3}$$

where $X_i = [x_1, \ldots, x_b]^T \in \mathbb{R}^{n\times D}$ is the data matrix that consists of n images and its corresponding noisy one-hot label matrix is $Y_i = [y_1, \cdots, y_n]^T \in \{0,1\}^{n\times C}$. X_j and Y_j are the randomly-shuffled versions of X_i and Y_i respectively. $\Lambda \in \mathbb{R}^{n\times n}$ is the diagonal matrix, whose ith diagonal element Λ_{ii} is randomly sampled from beta distribution $Beta(\alpha, \beta)$ with $\alpha = \beta$ and $\Lambda_{ii} \in [0, 1]$. And $\tilde{X}$ is the interpolated data matrix and $\tilde{Y}$ is the interpolated label matrix.

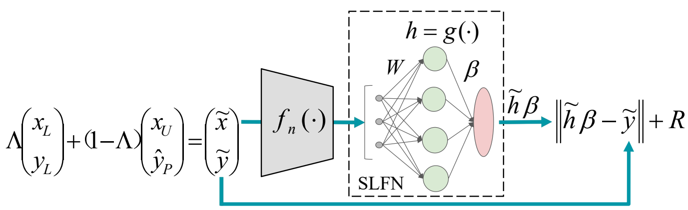


Figure 3: Schematic diagram of SLI-ELM processing one labeled sample (x_L, y_L) and one pseudo-labeled sample $(x_U, \hat{y}_p)$. SLFN is a single-layer feedforward neural network, which represents the basic structure of ELMs. R represents the squared Frobenius norm or ℓ_2 of β.

As shown in Fig.3, by introducing the stochastic linear interpolation to the ℓ_2-norm regularized ELM (Eq.(A.2)), we design the objective of SLI-ELM as Eq.(4).

$$\min_{\beta} \left\|\Lambda^{\frac{1}{2}}(\tilde{H}\beta - Y_i)\right\|_F^2 + \left\|(\boldsymbol{I} - \Lambda)^{\frac{1}{2}}(\tilde{H}\beta - Y_j)\right\|_F^2 + c\left\|\beta\right\|_F^2 \tag{4}$$

where $\tilde{H} = g\big(f_n\big(\tilde{X}\big)\big)$ is the hidden layer output matrix processed by $g\big(f_n\big(\cdot\big)\big)$ of DF-DAELM, F-norm is Frobenius norm, and c represents the coefficient of F-norm. The first two terms are the weighted losses that $\tilde{H}$ is classified as Y_i and Y_j respectively. And the two weighted losses are similar to the weighted least squares, where Λ and $\boldsymbol{I} - \Lambda$ are the weight diagonal matrix respectively. However, the first two terms also make the corresponding optimization problem hard and inefficient to solve. To tackle the problem, we propose the equivalent formula:

$$\min_{\beta} \left\| \tilde{H}\beta - \tilde{Y} \right\|_F^2 + c \left\| \beta \right\|_F^2 \tag{5}$$

Here, $\tilde{Y} = \Lambda Y_i + \big(\boldsymbol{I} - \Lambda\big)Y_j$ (the proof is given in appendix (P.1)). Through this equivalent formula, we can easily obtain the analytical solution and iterative algorithm. According to Eq.(5) and Eq.(A.3), the analytical solution is as follows:

$$\begin{aligned} \beta^* &= \big(\tilde{H}^T\tilde{H} + c\boldsymbol{I}\big)^{-1}\tilde{H}^T\tilde{Y} \quad if\ n \geq d, \\ \beta^* &= \tilde{H}^T\big(\tilde{H}\tilde{H}^T + c\boldsymbol{I}\big)^{-1}\tilde{Y} \quad other. \end{aligned} \tag{6}$$

In order to obtain an effective model, we also propose the stochastic linear interpolation online ELM (SLI-OELM), which can process data in batches (with fixed or varying size). Specifically, for any epoch, the k-th batch of data is defined as $\{X_i, Y_i\}_k$, and its corresponding shuffled batch is $\{X_j, Y_j\}_k$. After conducting stochastic linear interpolation Eq.(3), the interpolated k-th batch data is $\big\{\tilde{X}, \tilde{Y}\big\}_k$. Their corresponding hidden layer output matrix is $H_k = g\big(f\big(\tilde{X}_k\big)\big)$.

Based on Eq.(A.5) and the recursive least squares algorithm, Eq.(7) gives the initialization formula of SLI-OELM for the output weight β_0 and Eq.(8) provides the recursive formula of SLI-OELM for β^{k+1}. In general, SLI-OELM consists of two phases, namely an initialization phase and a recursive learning phase. Note that in the initialization phase, the number of data should be at least equal to the number of hidden nodes.

$$\begin{aligned} \beta_0 &= K_0^{-1}\tilde{H}_0^T\tilde{Y}_0 \\ K_0 &= (\tilde{H}_0^T\tilde{H}_0 + c\boldsymbol{I})^{-1} \end{aligned} \tag{7}$$

$$\begin{aligned} K_{k+1} &= K_k - K_k\tilde{H}_{k+1}^T\big(\boldsymbol{I} + \tilde{H}_{k+1}K_k\tilde{H}_{k+1}^T\big)^{-1}\tilde{H}_{k+1}K_k \\ \beta^{k+1} &= \beta^k + K_{k+1}\tilde{H}_{k+1}^T\big(\tilde{Y}_{k+1} - \tilde{H}_{k+1}\beta^k\big) \end{aligned} \tag{8}$$

Based on the above analysis, the SLI-OELM algorithm 1 for DF-DAELM can be summarized as follows.

Algorithm 1: DF-DAELM with SLI-OELM

Input: training dataset $\mathcal{D} = \left\{(x_i, y_i)\right\}_{i=1}^{n}$, trained deep feature representation model $f_n(\cdot)$, Beta distribution parameter α, the penalty coefficient c of F-norm, initialization batch size B_{ini} and iteration batch size B.

Output: the output weights β of SLI-OELM

Initialization phase:

Randomly generate hidden node parameters w, b of $g(\cdot)$

Sample $X, Y = \{(x, y)\}_i^{B_{ini}} \mathcal{D}$

$H_0 = g(f(X))$

Initialize K_0 and β^0 by using Eq.(3) and Eq.(7)

while *not converge* **do**

    for $t = 1, \cdots, T$ **do**

        Sample $X, Y = \{(x_i, y_i)\}_i^B \sim D(x, y)$

        Execute stochastic linear interpolation:

        $X_i, Y_i =$ shuffle(X, Y), $X_j, Y_j = X, Y$

        $\lambda \sim Beta(\alpha, \alpha)$

        $X_{k+1} = diag(\lambda) X_i + diag(1-\lambda) X_j$

        $Y_{k+1} = diag(\lambda) Y_i + diag(1-\lambda) Y_j$

        Calculate the hidden layer output matrix:

        $H_{k+1} = g(f(X_{k+1}))$

        Updating K_{k+1} and β^{k+1} by using Eq.(8)

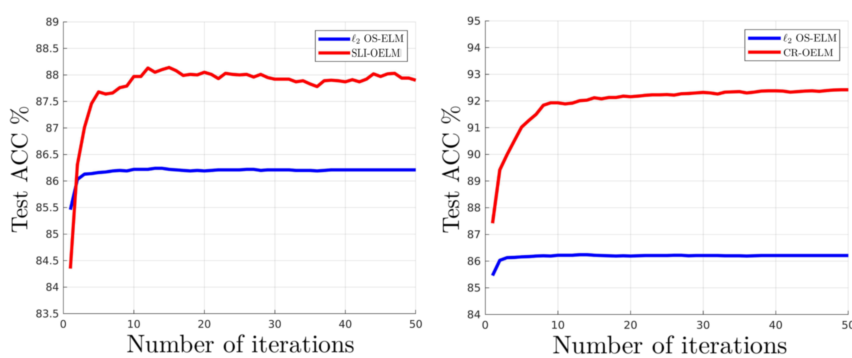
        Let $k \leftarrow k + 1$

    end

end

Remark 1. *Intuitively, only by combining incorrect labels and correct labels can the stochastic linear interpolation balance the huge penalty of incorrect labels to the decision hyperplane, so as to improve the robustness of the model. However,* *in fact, it is difficult to distinguish incorrect labels from all samples, so the proposed SLI-OELM randomly combines all samples indiscriminately, which is also proved to be effective by experiments with* $10K$ *samples at* 60% *noise level, as shown in Fig.6(a).*

Remark 2. *For the convergence of algorithm 1, the problem in Eq.(5) is a convex problem and similar to* ℓ_2*-norm regularized OS-ELM [55]. Meanwhile, from Eq.(7) and Eq.(8), it can be seen that the recursive implementation of the analytical solution (6) is similar to recursive least-squares method. Hence, all the convergence results of recursive least-squares (RLS) can be applied here [55]. Here, we define the complexity of a linear interpolation for a sample as* $O(z)$*. For the computational complexity, compared with basic* ℓ_2*-norm regularized OS-ELM, SLI-OELM only simply increases the cost of addition for each input sample pair and uses almost no additional computation, and its computational complexity is about* $t \cdot (O(n^3) + n \cdot O(z))$ *or* $t \cdot (O(d^3) + n \cdot O(z))$*, where* t *is the number of iterations. Moreover, empirical results show that the algorithm converges in less than* 15 *iterations, as shown in Fig.4(a), so only a few extra calculation is needed for training SLI-OELM.*



(a) Comparison of SLI-OELM and ℓ_2-norm regularized OS-ELM

(b) Comparison of CR-OELM and ℓ_2-norm regularized OS-ELM

Figure 4: Convergence curve on MNIST using 10000 training samples at 60 % noise-level

3.2.2. Consistent Regularization online ELM (CR-OELM)

In this section, in order to realize a noise-tolerant classifier learning, we further optimize the learning process of ELM based on the smoothness assumption and propose another novel online ELM classification algorithm named CR-OELM by introducing consistency regularization to the objective function of the traditional ELMs. Algorithm 2 gives the pseudocode description of CR-OELM.

CR-OELM is established on the smoothness assumption, that is, for a sample and its neighborhood, the prediction of the model should be the same. Concretely speaking, a classification model $F : \mathbb{R}^d \to \mathbb{R}$ with good generalization performance should satisfy $l-$Lipschitz continuity:

$$\|F(x_i) - F(x_j)\| <= l\|x_i - x_j\| = l\|\delta\| \tag{9}$$

where $l \in \mathbb{R}^+$, δ is a small amount, and for all $x_i \in \mathbb{R}^d$, $x_j = x_i + \delta$. $\|F(x_i) - F(x_j)\|$ is also called consistency regularization iterm [10, 8, 63, 10], which is able to reflect the conctent where the model $F(\cdot)$ has overfitted. Specifically, given a model $F(\cdot)$ fitted by a clean dataset, $\|F(x) - F(x+\delta)\| \approx 0$ for all $x \in \mathbb{R}^d$. And if there are some sparse noisy samples in the dataset and the model $F(\cdot)$ has already fitted them, for any one x of the noisy samples, $\|F(x) - F(x+\delta)\| > 0$. Therefore, this term can be used to indicate whether the model has overfitted the noisy samples.

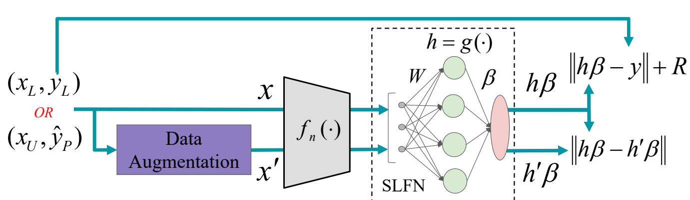


Figure 5: Schematic diagram of CR-ELM processing labeled samples (x_L, y_L) or pseudo-labeled samples $(x_U, \hat{y}_P)$. $\|h\beta - y\|$ and $\|h\beta - \acute{h}\beta\|$ are the main regularization terms of CR-ELM. SLFN is a single-layer feedforward neural network as the basic structure of ELMs.

Based on the above analysis, we proposed to introduce the consistency regularization into Eq:(A.2) to improve the classification model's noise-tolerant capability, as shown in Fig.5. The objective function of CR-ELM is formulated as shown in Eq.(10).

$$\min_{\beta} \|H\beta - Y\|_F^2 + c_0\|\beta\|_F^2 + c_1\|H\beta - \acute{H}\beta\|_F^2 \tag{10}$$

Here, we assume that $E(\cdot)$ is a perturbation function representing some data augmentation operation, such as random rotation, affine transformation or cropping, etc. For the data matrix $X = \{x_i\}_{i=1}^n$, its perturbed data matrix is $\acute{X} = E(X)$. Their corresponding hidden layer output matrix are $H \in \mathbb{R}^{n\times m}$ and $\acute{H} \in \mathbb{R}^{n\times m}$ respectively, processed by $g(f(\cdot)_n)$. In formula (10), $c_1\|H\beta - \acute{H}\beta\|_F^2$ is the consistency regularization term, c_1 is penalty coefficient of consistency regularization term.

Remark 3. *From the Eq.(11), we observe that when the consistency regularization term $\|H\beta - \acute{H}\beta\|_F^2$ becomes larger, $\big(H - \acute{H}\big)^T\big(H - \acute{H}\big)$ is larger. Thereby, the denominator of the analysis formula (Eq.(11)) is large, and the contribution of the corresponding samples to the output weights will be small in the end. Therefore, CR-ELM can implicitly adjust the output weight adaptively to reduce the risk of overfitting to incorrect labels. It is similar to the weighted loss function [39] or ℓ_{21}-norm ELMs [46], but it can implicitly detect and punish noisy samples but without complicated solution costs.*

The closed-form solution of CR-ELM can be calculated according to Eq.(11) (The derivation process can be found in appendix C.1).

$$\begin{aligned} \beta^* &= \big(H^T H + c_1\big(H - \acute{H}\big)^T\big(H - \acute{H}\big) + c_0\boldsymbol{I}\big)^{-1} H^T Y \\ \beta^* &= H^T\big(HH^T + c_1\big(H - \acute{H}\big)\big(H - \acute{H}\big)^T + c_0\boldsymbol{I}\big)^{-1} Y \end{aligned} \tag{11}$$

Furthermore, in order to make CR-ELM be able to online deal with data one by one or trunk by trunk, we propose the consistency regularization online ELM (CR-OELM) and come up with the recursive update formula below (The derivation process can be found in appendix C.2).

$$
\begin{aligned}
K_0 &= \big((1+c_1)H_0^T H_0 + c_1\big(\acute{H}_0^T \acute{H}_0 - 2H_0^T \acute{H}_0\big) + c_0\boldsymbol{I}\big) \\
&= H_0^T\big((1+c_1)H_0 - 2c_1\acute{H}_0\big) + c_1\acute{H}_0^T\acute{H}_0 + c_0\boldsymbol{I} \\
\beta_0 &= K_0^{-1}H_0^T Y_0
\end{aligned}
\tag{12}
$$

$$
\begin{aligned}
K_{k+1} &= K_k + H_{k+1}^T\big((1+c_1)H_{k+1} - 2c_1\acute{H}_{k+1}\big) + c_1\acute{H}_{k+1}^T\acute{H}_{k+1} \\
\beta_{k+1} &= \beta_k + K_{k+1}^{-1}\big(H_{k+1}^T Y_{k+1} - \big(H_{k+1}^T\big((1+c_1)H_{k+1} - 2c_1\acute{H}_{k+1}\big) \\
&\qquad + c_1\acute{H}_{k+1}^T\acute{H}_{k+1}\big)\beta_k\big)
\end{aligned}
\tag{13}
$$

Remark 4. *The consistency regularization can be interpreted as the approximate manifold regularization [54]. It is worth noting that CR-OELM constrains the manifold structure of the model through data augmentation rather than the Graph-Laplace constraint calculated in advance [30, 17]. Specifically, the consistency regularization loss implicitly penalizes input-output Jacobian norm* $\lim_{\delta\to 0}\frac{1}{\delta^2}\frac{1}{n}\sum_{i=1}^{n}\|\beta g(f(x_i+\delta)) - \beta g(f(x_i))\|_F^2 \approx \mathbb{E}_x[\|J_x\|_F^2]$*, where* J_x *is the jacobian of outputs of* $g(\cdot)$ *with respect to its inputs evaluated at sample point* x*. Given that data augmentation* $\delta = E(x)$ *can be viewed as approximating elemtent of the tangent space* $T_x(\mathcal{M})$ *at any sample* x*,* $\mathbb{E}_x[\|J_x\|_F^2]$ *is equivalent to manifold regularization* $\|\nabla J_{\mathcal{M}}\|_F^2$*.*

Remark 5. *The computational complexity of Algorithm 2 is determined by the iterative number* t *and the computational cost in one iteration. We mainly analyze the latter. Firstly, since there are multiple data augmentation methods, we uniformly define the complexity of performing a data augmentation operation for a sample* $O(z)$*. Therefore, the complexity of data augmentation in one iteration is* $nO(z)$*. The computational complexity is* $O(n^3)$ *or* $O(d^3)$ *for the inverse of the matrix with size of* $n\times n$ *or* $d\times d$*. The computational complexity of the consistency regularization term is* $d*n^2$ *or* $n*d^2$*. So the computational complexity of Algorithm 2 is about* $t\cdot(O(n^3)+n\cdot O(z))$ *or* $t\cdot(O(d^3)+n\cdot O(z))$*. As for the iterative number, the empirical results show that the algorithm converges in less than* 10 *iterations, as shown in Fig.4(b).*

Algorithm 2: DF-DAELM with CR-OELM

Input: training dataset $\mathcal{D} = \left\{(x_i, y_i)\right\}_{i=1}^{n}$, trained deep feature representation model $f_n(\cdot)$, the penalty coefficient c_0 of F-norm, the penalty coefficient c_1 of consistency regularization, the perturbation function $E(\cdot)$, initialization batch size B_{ini} and iteration batch size B.
Output: the output weights β of CR-OELM
Initialization phase:
Randomly generate hidden node parameters w, b of $g(\cdot)$
Sample $X, Y = \{(x, y)\}_i^{B_{ini}} \sim \mathcal{D}$
Generate neighbor samples $\acute{X} = E(X)$
$H_0 = g\big(f(X)\big)$, $\acute{H}_0 = g\big(f(\acute{X})\big)$
Initialize K_0 and β^0 by using Eq.(12)
while *not converge* **do**
  for $t = 1, \cdots, T$ **do**
    Sample $X, Y = \{(x_i, y_i)\}_i^{B} \sim \mathcal{D}$
    Generate neighbor samples:
    $\acute{X} = E(X)$
    Calculate the hidden layer output matrix:
    $H_{k+1} = g\big(f(X)\big)$, $\acute{H}_{k+1} = g\big(f(\acute{X})\big)$
    Updating K_{k+1} and β^{k+1} by using Eq.(13)
    Let $k \leftarrow k + 1$
  end
end

Altogether, compared with the existing Deep SSL methods in references [13, 10, 18, 5, 17], the proposed DF-DAELM not only maintains the consistency between feature representation and classification task but also eliminates the intractable confirmation bias problem by retraining the classifier. Through data augmentation, the proposed two classifiers minimize the vicinal risk to reduce the dependence of the previous robust ELMs on regularization [46, 47], and can automatically explore the knowledge of the data itself instead of empirical knowledge like [49, 50] to improve noise robustness. Meanwhile, the proposed two data-augmented ELMs are very convenient to be integrated with deep neural networks and effectively process high-dimensional data. Based on these two ELMs, the proposed DF-DAELM can improve the performance without the help of kernel methods [34], multi-view [17, 42] or manifold regularization [30].

4. Experiments and discussions

In this section, we evaluate the proposed DF-DAELM algorithm on several SSL benchmark datasets. In section 4.2, we perform comparative experiments with several popular pre-training-based SSL approaches and provide an extensive ablation study to explore and analyze the effectiveness of various components. Section 4.3 conducts several experiments to verify the effectiveness of the proposed two noise-robust classifiers (SLI-OELM and CR-OELM) for DF-DAELM on the modified MNIST dataset. Finally, We demonstrate the proposed DF-DAELM with multiple state-of-the-art (SOTA) SSL methods in section 4.4.

4.1. Dataset

We assess the proposed method on 3 SSL benchmark datasets: CIFAR-10, CIFAR-100 [64] and SVHN [65]. For CIFAR-10/100, these datasets contain 10 and 100 classes respectively with $50K$ RGB images for training and $10K$ for testing. SVHN contains of 73257 training samples and 26032 test samples. The resolution of the sample images in SVHN is 32×32, which also has 10 different classes. And each example is a close-up of a house number and the class represents the identity of the digit at the center of the image.

We evaluate the proposed two robust classifiers (SLI-OELM and CR-OELM) of DF-DAELM on MNIST dataset. It is a standard dataset for handwritten digit classification tasks, which includes $70K$ 28×28 sample images.

4.2. Comparative experiment with pre-training-based methods and Ablation study

In this section, we perform comparative experiments with several popular pre-training-based SSL approaches and provided an extensive ablation study. We perform experiments on CIFAR-10/100 of 45k samples (4k labeled samples included). The original training dataset is randomly split into a training subdataset of $41K$ samples with $4K$ labeled samples and a validation subdataset with $5K$ samples. For the fairness of comparison, each experiment is executed in the same training, validation. And the error rate on the test dataset is reported. Meanwhile, all experiments use PreAct ResNet-18 (PR-18) backbone [57].

4.2.1. Implementation Details

In the training process of the deep feature learning, following [18], we adopted SGD with momentum of 0.9, weight decay of 10^{-4} and batch size of 100. Training always started with a relatively high learning rate 0.1. We trained 400 epochs (reducing learning rate to 0.01 and 0.001 in epochs 250 and 350 respectively) and used 10 epoch warm-up with labeled data for CIFAR-10/100. Weight normalization [66] was used in all networks. Following [18], Mixup [21] was adopted with $\alpha = 4$. For data augmentation, we randomly augmented images using a reflect padding, a color jitter, random crop and a random horizontal flip. We then normalized images to have channel-wise zero mean and unit variance over training data.

In the training process of SLI-OELM and CR-OELM, the methods of data augmentation were the same as the one that was adopted for training deep feature learning. The dropout was also used as the structural perturbation of deep feature model with the fixed trained parameter. Note that, to facilitate the data augmentation of SLI-OELM and CR-OELM, the features fed into ELM classifier were directly inferred by the neural network instead of the processed feature matrix of all data. The classifier was trained for up to 50 epochs, and the one that showed the best accuracy was selected. According to the hyperparameter analysis (Section 4.5), for SLI-OELM, the coefficient c_0 of Frobenius norm and the coefficient of the Beta distribution α were set as 0.01 and 6 respectively. As for CR-OELM, the coefficient c_0 was the same as that of SLI-OELM, and the coefficient c_1 of consistency regularization term was set to 0.42. The activation function adopts LeakyReLU. Experiments were conducted in PyTorch environment with 2 NVIDIA 2080 Ti GPUs.

4.2.2. Comparison methods

To show the superiority of DF-DAELM, we adopt a fully supervised method and several popular pre-training-based semi-supervised classification methods: Non-Parametric Instance Discrimination [27] + fine-tune (U+fine-tune), VAE [26]+fine-tune (V+fine-tune), Non-Parametric Instance Discrimination [27]+ SSELM [29] (U+SSELM), and VAE [26]+SSELM [29] (V+SSELM). Like the previous works [10, 11], we use deep convolutional features+softmax as the fully supervised method, which is only trained on the same labeled data as the semi-supervised method. Meanwhile, we study the effect of the various components of DF-DAELM to verify their contributions. Specifically, in order to verify the proposed two data augmented classifiers, we adopt two fully supervised classifiers to combine with deep convolutional features of DF-DAELM, i.e. CNN of DF-DAELM+softmax (CNN without ELMs) and CNN of DF-DAELM with ℓ_2 OS-ELM (CNN with ℓ_2 OS-ELM). Among them, CNN without ELMs is also a plain pseudo-label-based SSL method. In order to address the effectiveness of feature fusion, the proposed methods (DF-DAELM with SLI-OELM or CR-OELM) based on high-level semantic features (Single-) or multi-level features (Multi-$*$) are compared respectively. Meanwhile, the number of channels in the first two layers of the PR-18 network is small, which has very little useful information after GAP [62], so in the feature fusion experiment, we only compared the features of the last three layers.

4.2.3. Discussion and Analysis

Table 1 shows the test error of multiple comparison experiments on CIFAR10/100 with 4000 labels.

In terms of pre-training-based semi-supervised classification methods, the proposed DF-DAELM has a larger performance improvement than the 4 pre-training-based semi-supervised classification methods (U/V+SSELM/fine-tune). Although these 4 methods perform well on unsupervised problems, when combined with specific classification tasks, their performance improvements are trivial or even suffer from a worse result compared with the fully supervised method. Especially when they are combined with SSELM (U+SSELM, V+SSELM), the performance dropped a lot. The reason for this phenomenon is the inconsistency between the feature representation obtained by the unsupervised method and the ultimate classification task. This result also supports our view from the side, that is, the introduction of label information in the feature learning stage will improve the performance of the model, such as the feature representation learning method used by our method (DF-DAELM) in the first stage.

In terms of the two regularization terms (R_0 and R_1 of Eq.(2)) of the self-training SSL method used by our DF-DAELM, we directly use the hyperparameters provided by [58] to constrain the self-training semi-supervised model and set λ_1 and λ_2 to 0.4 and 0.8 respectively. We just conduct a simple ablation study with or without the two hyperparameters as shown in Table 3. The results verify that the combination of the two regularizations is important to improve the overall performance of the feature representation model.

In terms of the two proposed robust ELMs, the test errors of (Single-CNN+SLI-OELM(ours)) and (Single-CNN+CR-OELM(ours)) are total lower than the plain pseudo-label-based method (CNN without ELMs). Meanwhile, (Single-CNN+SLI-OELM(ours)) and (Single-CNN+CR-OELM(ours)) are almost lower than (CNN+ℓ_2 OS-ELM)), which verified that the two proposed SLI-OELM and CR-OELM can improve the robustness of traditional OS-ELM.

In terms of feature fusion, the result exhibits that the combination of the proposed data augmented ELMs and features fusion has a better anti-noise performance. It is worth noting that not fusing any layer features with the last layer features could improve the performance. Because, in deep neural networks, the shallowest features are less affected by noisy labels due to their long distance from the label, while they usually only contain some local and basic information. Thus the discrimination of the shallowest features is usually poor. Meanwhile, the deeper features contain semantic information but are susceptible to noise interference because they are closer to the noisy labels. So, according to this inference, for the PR-18 network with 5 layers used in Table 1, its performance of the fusion between the middle layers and the last layer could be better. Our experiments have also verified this point, namely, the performance of fusion between the third feature layer and the last layer is better.

Table 1: Comparison with baselines and ablation study. All values are error rates on CIFAR-10/100 with 4000 labels. For multi-$*$, $*$ represents the fusion between the features of the last layer and the features of $*$-th layer from last. Single- represents the last layer of features as the input of ELM.

Method	CIFAR10	CIFAR100
	4000	4000
Fully supervised	28.56	70.58
U[27]+fine-tune	28.57	70.59
V[26]+fine-tune	31.52	75.31
U[27]+SSELM[29]	63.62	89.17
V[26]+SSELM[29]	64.28	90.43
CNN without ELMs	10.36	48.30
CNN+ℓ_2 OS-ELM [55]	10.23	47.05
Single-CNN+SLI-OELM (ours)	10.12	47.22
Multi-2-CNN+SLI-OELM(ours)	10.18	47.10
Multi-3-CNN+SLI-OELM(ours)	**10.09**	**46.76**
Single-CNN+CR-OELM(ours)	**9.96**	46.64
Multi-2-CNN+CR-OELM(ours)	10.08	46.39
Multi-3-CNN+CR-OELM(ours)	9.97	**45.80**

At last, taking the performance of the pseudo-label-based method (CNN without ELMs) as the baseline, the average improvement rates of performance of SLI-OELM and CR-OELM on CIFAR-10 are 2.48% and 2.7% respectively, while the average improvement rates of SLI-OELM and CR-OELM on CIFAR-100 are 3.4% and 4.19%, respectively. These results indicate that the two methods have a higher performance improvement on CIFAR-100 than on CIFAR-10, marking that our method is more suitable for classification scenarios with insufficient sample size. Meanwhile, it also shows that the anti-noise ability of CR-OELM based on data augmentation is better than that of SLI-OELM. In order to further explore the characteristics of the proposed SLI-OELM and CR-OELM, we conducted a detailed study in the next section.

4.3. Robustness Experiments for SLI-OELM and CR-OELM with different label noise levels and data scales

To study the advancement of the proposed two robust ELMs (SLI-OELM and CR-OELM) with different label noise levels and data scales, we conducted a variety of comparative experiments on MNIST.

Since SLI-OELM and CR-OELM play the role of a fully-supervised classifier with noise-tolerant in the proposed DF-DAELM framework, this section only studies their performance under supervision. According to [67] and the statistics of the proportion of noise in the pseudo labels generated in the first stage (see Fig.2(b)), we conducted the experiment with symmetric label noise, which is generated by randomly replacing the labels for a percentage of the training data with all possible labels. Specifically, we added 20%, 40%, 60%, 70% and 80% symmetrical noise to the total labels. At the same time, we changed the scale of the 50K training samples: 100%, 50%, 10%, 1%, 0.1%. Three robust ELMs (ELM with ℓ_2-norm (ℓ_2 OS-ELM) [55], Random Fourier ELM with ℓ_{21}-norm regularization (RFELM) [46], Orthogonal ELM (Orth-ELM) [47]) are used for comparison. As for the three robust ELM methods, a single hidden layer with 2500 random neurons is used, and LeakyRelu is used as the activation function.

As shown in Fig.6, each subfigure represents the performance curves of different methods with different scales under a certain noise level. In the case of low noise ratio ($\leq 20\%$) and larger data scale ($50K$), ℓ_2 OS-ELM, RFELM and Orth-ELM are comparable, as shown in the enlarged part in Fig.6(a) 6(b) and 6(c). But in the case of large label noise rate ($40\% \sim 80\%$) and smaller data scale ($\leq 20K$), the performance of SLI-OELM and CR-ELM is more prominent. The main reason is that our proposed methods use an iterative batching technique based on data augmentation, which increases the diversity of the samples and therefore improves the performance in the case of a small data scale. These results demonstrate that the performances of our proposed methods do surpass methods based on regularization (ℓ_2 OS-ELM and Orth-ELM) or noisy sample weighting (RFELM) in most cases. We also have found that CR-OELM has stronger anti-noise ability, while SLI-ELM is weaker. The reason behind this phenomenon is that the regularization based on data augmentation can effectively detect and punish noisy samples, while SLI-OELM can only correct wrong samples through random combinations of other noise-free samples and has a weaker ability to detect and punish noisy samples.

Table 2: Average runtime of multiple experiments with different label noise levels and data scales on MNIST.

ℓ_2 OS-ELM	SLI-OELM	CR-OELM
38.6612 s	232.8669 s	145.9241 s

In terms of efficiency, as shown in Table 2, the runtime of SLI-OELM is the most expensive, followed by CR-OELM. However, according to Remark 2 and 5, the one-time calculation cost of SLI-OELM is the same as that of CR-OELM. For specific calculations, as for the former, the one-time calculation of the inverse of the matrices is $M_{SLI} = (HH^T + c\mathbf{I})$, which is the same as ℓ_2 OS-ELM. The latter is $M_{CR} = (HH^T + c_1(H - \acute{H})(H - \acute{H})^T + c_0\mathbf{I})$. Due to the same size of the M_{SLI} and M_{CR} matrices when the input data is the same, the inverse cost of the matrix M_{SLI} and M_{CR} is linear, and the cost of M_{CR} is slightly higher. So the one-time calculation cost of SLI-OELM is less than that of CR-OELM. Based on the above inference, the results in Table 2 are explainable because stochastic linear interpolation could cause the model to be unstable, SLI-OELM will take more iterations to reach the convergence state, which is supported by Fig.4(a)(which shows a larger fluctuation range of the red convergence curve of SLI-OEM than that of CR-OELM). Finally, with the aid of data augmentation, the proposed SLI-OELM and CR-OELM have a simple solution that is similar to that of the standard ℓ_2 OS-ELM, which can be easily solved iteratively. These two algorithms converge in less than 30 iterations as shown in Fig.4.

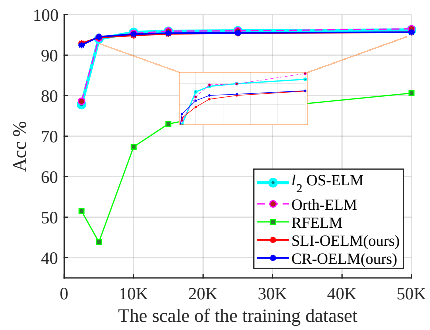
Table 3: The ablation study of R_0 and R_1. All values are error rates on CIFAR-10/100 with 4000 labels.

CNN of DF-DAELM		CIFAR10	CIFAR100
R_0	R_1	4000	4000
	✓	11.83	88.63
✓		23.24	67.98
		19.62	67.49
✓	✓	**10.36**	**47.39**

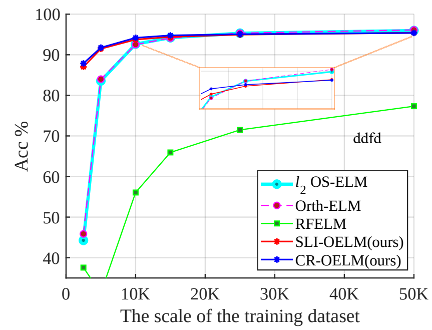
4.4. Comparison with the state-of-the-art one-stage methods

In this section, we compared the proposed DF-DAELM with multiple related state-of-the-art (SOTA) SSL methods. For the training process of deep neural networks, following Section 4.2.1, we experimented with 13-CNN (3M) [54] and Wide-ResNet-28-2(WR-28-2) (1M) [66] to study the generalization ability of the proposed method. The experiments were carried out on CIFAR10/100 dataset. Following [13, 18], we randomly sampled 500, 1000, and 4000 labels for CIFAR-10 while 4000 and 10000 labels for CIFAR-100. We created 4 splits for each number of labeled samples with different random seeds respectively. And the error rates were calculated by the mean and variance across splits.

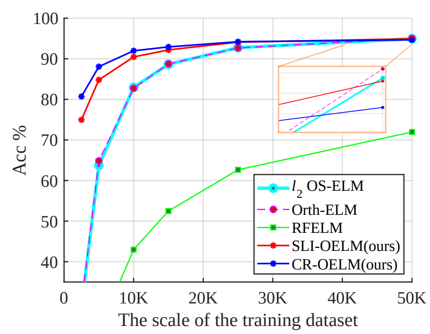
!



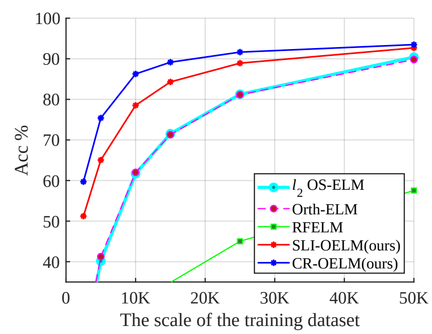
(a) Label noise level: 0%



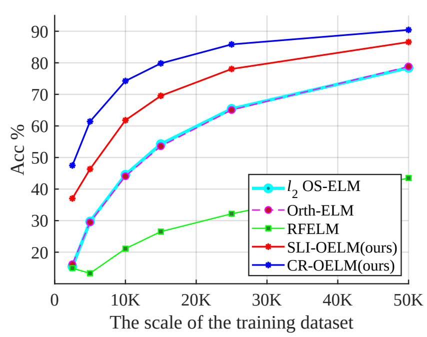
(b) Label noise level: 20%



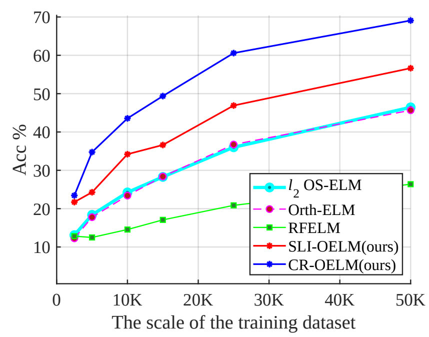
(c) Label noise level: 40%



(d) Label noise level: 60%



(e) Label noise level: 70%



(f) Label noise level: 80%

Figure 6: Robustness Experiments with different label noise levels and data scales

In order to show the superiority of the proposed DF-DAELM framework, we choose 11 representatives state-of-the-art methods for comparison. As is shown in Table 4 and Table 6, we compared our proposed method with Π model [9], Temporal Ensemble [9], Mean Teacher [10], VAT [8], MT-fast-SWA [54], SING [11], LP [5], ICT [12], MixMatch [13], WCP [7], NS_3L [22]. Results of the compared methods are taken from existing literatures.

Table 4: Test error on CIFAR-10/100 for the proposed method using the 13-CNN network.

Method	CIFAR-10			CIFAR-100	
	500	1000	4000	4000	10000
Π model [9]	-	31.65±1.20	12.36±0.31	-	39.19±0.36
Temporal Ensemble [9]	-	23.31±1.01	12.16±0.24	-	38.65±0.51
Mean Teacher [10]	27.45±2.64	21.55±1.48	12.31±0.28	45.36±0.49	36.08±0.51
Temporal Ensemble+SING [11]	-	18.41±0.52	10.93±0.14	-	-
MT-fast-SWA [54]	-	15.58	9.05	-	-
LP [5]	32.40±1.80	22.02±0.88	12.69±0.29	46.20±0.76	38.43±1.88
ICT [12]	-	15.48±0.78	7.29±0.0.02	-	-
WCP [7]	-	17.62±1.52	9.27±0.31	-	-
Mean Teacher+LP [5]	24.02±2.44	16.93±0.70	10.61±0.28	43.73±0.20	35.92±0.47
SLI-OELM(Ours)	9.04±0.30	7.75±0.02	6.60±0.01	42.27±0.21	36.67±0.31
SLI-OELM with dropout(Ours)	9.15±0.21	7.90±0.02	6.64±0.01	41.72±0.29	35.93±0.39
CR-OELM(Ours)	9.51±0.37	7.81±0.06	6.24±0.01	40.48±0.34	34.73±0.23
CR-OELM with dropout(Ours)	9.07±0.38	7.57±0.02	6.17±0.01	40.24±0.24	34.47±0.24

Table 4 and Table 6 show the test error of different methods on CIFAR-10/100 with 13-CNN network [54] or WR-28-2 network [66]. The red, green, and blue fonts indicate the top three methods. For 13-CNN network structure, as shown in Table 4, the proposed method obtained the best resluts under various proportions of labeled samples. For WR-28-2 network, as shown in Table 6, although our method cannot surpass MixMatch [13] in some cases, it's performance still occupies the top two, and the biggest gap when compared to MixMatch is less than 0.8%.

In terms of the generalization and transferability of DF-DAELM, SLI-OELM, and CR-OELM, we replaced the feature representation model adopted by DF-DAELM with two SOTA methods: MixMatch[13] and FixMatch[24]. We reproduced their methods based on [13, 24]. Here, we did not use the teacher-student model [10] but a single model, which is based on the backbone network Wide-ResNet-28-2(WR-28-2). Then, these two baseline models were trained for 90 epochs on the SVHN benchmark and 250 epochs on the CIFAR-10 benchmark. Other experimental settings are based on [13, 24]. For the hyperparameters in our method, we used the parameter values given in Section 4.5. As shown in Table 5, the results show that the performance of SLI-OELM and CR-OELM is greater than that of the original model, which has verified that our proposed DF-DAELM is a general deep semi-supervised classifier.

Table 5: Test errors achieved by MixMatch [13]/FixMatch [24] and MixMatch/FixMatch+SLI-OELM/CR-OELM(our) on the standard benchmark of CIFAR-10 and SVHN with all but 500 labels removed and all but 1,000 labels removed respectively. † means to reproduce the method.

Method	CIFAR-10		SVHN	
	500	1000	500	1000
MixMatch†	19.23±1.70	16.05±0.61	9.80±1.73	8.91±0.86
MixMatch†+SLI-OELM(our)	18.05±0.90	14.92±0.73	9.05±1.35	**7.78±0.57**
MixMatch†+CR-OELM(our)	**17.85±0.99**	**14.76±0.37**	**9.01±1.20**	7.99±0.60
FixMatch†	10.39±0.40	8.10±0.26	5.10±0.82	4.59±0.56
FixMatch†+SLI-OELM(our)	**9.10±0.19**	7.74±0.17	**4.70±0.38**	**4.40±0.49**
FixMatch†+CR-OELM(our)	9.33±0.14	**7.67±0.12**	4.77±0.50	4.50±0.49

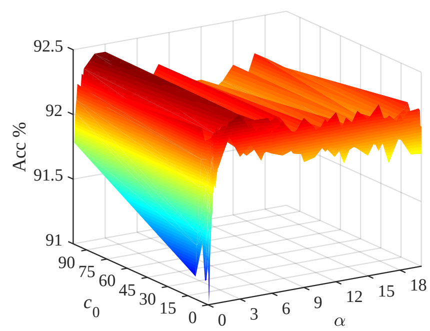
4.5. Hyperparameter Sensitivity

Firstly, we experimented on MNIST to explore the impact of the hyperparameters of SLI-OELM and CR-OELM on the classification accuracy. We split MNIST into a training dataset of $50K$ samples, a validation dataset of $10K$ samples, and a test dataset of $10K$ samples. In SLI-OELM, we vary c_0 from 10^{-4} to 10^{2} under each fixed α. Similarly, the coefficient α of SLI-OELM is finely tuned from 10^{-2} to 20 under each fixed c_0. As for CR-OELM, the value range of c_0 is the same as that of SLI-OELM and the coefficient c_1 is changed from 10^{-4} to 2. During this experiment, the number of neurons in the hidden layer of ELM was fixed at 2500.

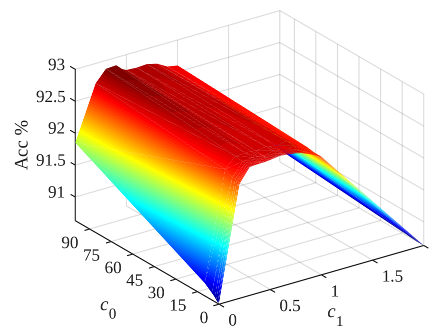
Table 6: Test error in CIFAR-10/100 for the proposed method using the WR-28-2 network.

Method	CIFAR-10			CIFAR-100
	500	1000	4000	10000
Π model[13]	-	-	14.01±0.38	37.88±0.11
Mean Teacher[13]	42.01±5.86	17.32±4.00	10.36±0.25	
VAT	26.11±1.52	18.68±0.40	11.05±0.31	44.38±0.56
MixMatch[13]	9.65±0.94	7.75±0.32	6.24±0.06	-
NS_3L[22]	-	-	16.03±0.05	46.34±0.37
VAT+NS_3L[22]	-	-	13.94±0.10	43.70±0.19
ICT[12]	42.33±0.08	-	7.66±0.07	-
SLI-OELM(Ours)	10.74±0.94	8.19±0.43	7.14±0.29	39.18±0.34
SLI-OELM with dropout(Ours)	10.58±0.99	8.07±0.62	7.16±0.35	38.47±0.13
CR-OELM(Ours)	10.50±0.81	7.62±0.65	6.79±0.68	36.64±0.08
CR-OELM with dropout(Ours)	10.45±0.96	8.23±0.15	6.52±0.04	36.52±0.05

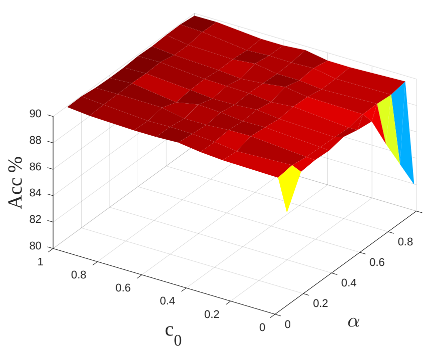
Fig.7(b) and Fig.7(a) show the performance of c_0, α and c_1 on the validation dataset. From these two figures, we can observe that the curve of classification of SLI-OELM on clean dataset firstly goes up as the increase of parameter α independent of c_0. When α is equal to 0.6, we can get the optimal values. We can also see that the performance of CR-OELM is related to c_1, but not to c_0. The optimal value is obtained at $C_1 = 0.42$. As for the CIFAR-10, we explored the impact of the hyperparameters of SLI-OELM and CR-OELM based on the experimental settings in Subection 4.2, Fig.7(c) and Fig.7(d) show the performance of c_0, α and c_1. Through Fig.7(c) and 7(d), we can see that the optimal parameter values of both methods are different from those on the clean dataset, that is, α and c_1 are all related to c_0. Hence, after this experiment, we set the parameter c_0, c_1 and α to 10^{-2}, 0.42 and 0.6, respectively.



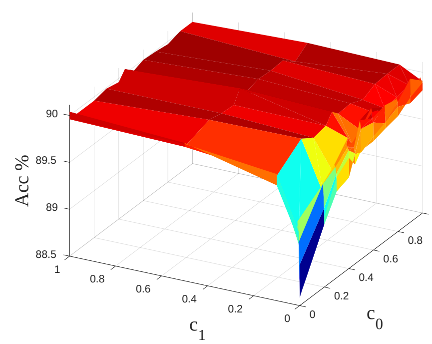
(a)



(b)



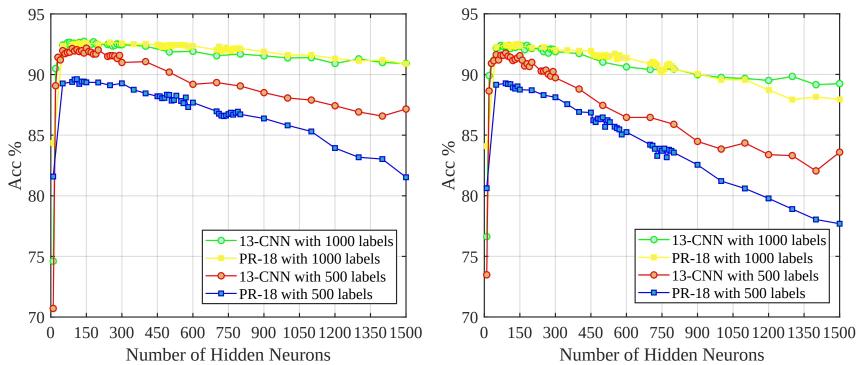
(c)



(d)

Figure 7: Evaluation results with different weights: (a) is validation accuracy (Acc) of SLI-OELM across c_0 and α on MNIST. (b) is validation accuracy of CR-OELM across c_0 and c_1 on MNIST. (c) is validation accuracy (Acc) of SLI-OELM across c_0 and α on CIFAR-10 with . (d) is validation accuracy (Acc) of SLI-OELM across c_0 and c_1 on CIFAR-10. (a)(b) are experiments conducted on the clean dataset, while (c)(d) are experiments conducted on extracted deep features and inferred pseudo labels with 4000 labeled samples.

In order to get the optimal number of hidden neurons, the validation experiments of SLI-OELM and CR-OELM were conducted across several networks based on deep features and pseudo labels generated in CIFAR-10 under 1000/500 samples. We split the original training dataset of CIFAR-10 into a smaller training dataset of $45K$ samples and a validation dataset of $5K$ samples. As shown in Fig.8, the best number of hidden layer neurons is roughly between 100 and 350.



(a) SLI-OELM (b) CR-OELM

Figure 8: Validation accuracy curve with the number of hidden neurons of SLI-OELM/CR-OELM on CIFAR-10 (500/1000 labels). All experiments were performed under fixed c_0, c_1 and α, where PR-18 and 13-CNN represent PreAct ResNet-18 and 13-CNN networks respectively.

In short, the above experiments and analysis verify the effectiveness of our proposed DF-DAELM in solving the problem of confirmation bias encountered by current deep SSL methods. Since DF-DAELM is a general deep SSL method, it could be used in a variety of scenarios with high annotation costs, such as medical diagnosis, hyperspectral images, traffic scene recognition in unmanned driving, 3D object detection in manipulator operation, and so on. Specifically, in medical diagnosis, due to the high similarity of data, many samples are difficult to manually annotate. This is an intractable issue for supervised models that require a large amount of labeled data. Fortunately, our proposed DF-DAELM is able to automatically use these unlabeled data to improve performance and reduce manual labeling costs. However, in the application process, it is important to note that the domain-specific step is to design the corresponding network structure according to different types of data, for example, using PointNet [68] or 3D convolutional network backbone to process lidar data. Finally, since DF-DAELM currently only focuses on classification problems, other issues need to be considered when migrated to other fields, such as the regression problem of the bounding box in object detection.

5. Conclusion

In this paper, we propose a robust semi-supervised classification approach (DF-DAELM) to solve the confirmation bias issue encountered by the pseudo-label-based semi-supervised methods. Specifically, based on the deep features and pseudo labels generated by semi-supervised pre-training, DF-DAELM designs two noise-robust classifiers (SLI-OELM and CR-OELM) to further improve the performance of the model. SLI-OELM firstly conducts stochastic linear interpolation to augment the data and then uses them to train extreme learning machines, which significantly strengthens the robustness of classification. And CR-OELM utilizes a consistency regularization term to constrain the parameter space of the ELM classifier, so that CR-OELM can implicitly detect and penalize the samples with noisy labels, preventing the ELM classifier from overfitting. For the computational complexity, the overhead of the proposed two data augmented ELMs is about $t \cdot (O(n^3) + n \cdot O(z))$ or $t \cdot (O(d^3) + n \cdot O(z))$, which is similar to standard OS-ELM [55, 38] but with an additional cost $t \cdot n \cdot O(z)$ for data augmentation operations. Comprehensive experiments demonstrate that DF-DAELM achieves competitive or even better performance on CIFAR-10/100 and SVHN over the related state-of-the-art methods. Meanwhile, for the proposed classifiers, experimental results on the MNIST dataset with different noise levels and sample scales demonstrate their superior performance, especially when the sample scale is small ($\leq 20K$) and the noise is strong ($40\% \sim 80\%$). In other words, exploiting the non-convex squared loss function can indeed help improve the robustness of the SSL algorithm.

However, some limitations of the proposed DF-DAELM still exist, such as, there is no further analysis and demonstration for the proposed multi-feature fusion to eliminate noisy features and the proposed data augmented ELMs are only applied to the mean square error (MSE) criterion. In the future work, we intend to extend the proposed DF-DAELM in three aspects: (1) Studying the rapid training strategy of the deep feature networks adpoted by DF-DAELM. (2) Studying the feature representation model based on the attention mechanism that can be dynamically updated, so that the proposed SLI-OELM and CR-OELM can not only punish noisy samples, but also update the network structure and parameters of the feature representation model adopted by DF-DAELM. (3) Extending our proposed data augmented ELMs (SLI-OELM and CR-OELM) to non-convex [49, 45, 50] and other methods based on special loss functions.

6. Acknowledgments

The work was supported by the National Natural Science Foundation of China (Grants Nos. 61825305, 62006237, 62022091). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

7. References

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, IEEE Computer Society, 2016, pp. 770–778. `doi:10.1109/CVPR.2016.90`.
URL `https://doi.org/10.1109/CVPR.2016.90`

[2] T. Chen, S. Kornblith, M. Norouzi, G. E. Hinton, A simple framework for contrastive learning of visual representations, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, Vol. 119 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 1597–1607.
URL `http://proceedings.mlr.press/v119/chen20j.html`

[3] G. E. Hinton, S. Osindero, Y. W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554. `doi:10.1162/neco`

.2006.18.7.1527.
URL https://doi.org/10.1162/neco.2006.18.7.1527

[4] D.-H. Lee, Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks, 2013.

[5] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 5070–5079. doi:10.1109/CVPR.2019.00521.
URL http://openaccess.thecvf.com/content_CVPR_2019/html/Iscen_Label_Propagation_for_Deep_Semi-Supervised_Learning_CVPR_2019_paper.html

[6] Y. Grandvalet, Y. Bengio, Semi-supervised learning by entropy minimization, in: Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada], 2004, pp. 529–536.
URL http://papers.nips.cc/paper/2740-semi-supervised-learning-by-entropy-minimization

[7] L. Zhang, G.-J. Qi, Wcp: Worst-case perturbations for semi-supervised deep learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3912–3921.

[8] T. Miyato, S. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: A regularization method for supervised and semi-supervised learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (8) (2019) 1979–1993.

[9] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net,

2017.
URL https://openreview.net/forum?id=BJ6oOfqge

[10] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017, pp. 1195–1204.
URL http://papers.nips.cc/paper/6719-mean-teachers-are-better-role-models-weight-averaged-consistency-targets-improve-semi-supervised-deep-learning-results

[11] Y. Luo, J. Zhu, M. Li, Y. Ren, B. Zhang, Smooth neighbors on teacher graphs for semi-supervised learning, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, IEEE Computer Society, 2018, pp. 8896–8905. doi:10.1109/CVPR.2018.00927.
URL http://openaccess.thecvf.com/content_cvpr_2018/html/Luo_Smooth_Neighbors_on_CVPR_2018_paper.html

[12] V. Verma, A. Lamb, J. Kannala, Y. Bengio, D. Lopez-Paz, Interpolation consistency training for semi-supervised learning, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 3635–3641. doi:10.24963/ijcai.2019/504.
URL https://doi.org/10.24963/ijcai.2019/504

[13] D. Berthelot, N. Carlini, I. J. Goodfellow, N. Papernot, A. Oliver, C. Raffel, Mixmatch: A holistic approach to semi-supervised learning, in: H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32: An-

nual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, 2019, pp. 5050–5060. URL `http://papers.nips.cc/paper/8749-mixmatch-a-holistic-approach-to-semi-supervised-learning`

[14] L. Yao, Z. Ge, Deep learning of semi supervised process data with hierarchical extreme learning machine and soft sensor application, IEEE Transactions on Industrial Electronics 65 (2) (2018) 1490–1498.

[15] P. Chang, J. Zhang, J. Hu, Z. Song, A deep neural network based on ELM for semi-supervised learning of image classification, Neural Process. Lett. 48 (1) (2018) 375–388. `doi:10.1007/s11063-017-9709-0`.
URL `https://doi.org/10.1007/s11063-017-9709-0`

[16] M. D. Tissera, M. D. McDonnell, Deep extreme learning machines: supervised autoencoding architecture for classification, Neurocomputing 174 (2016) 42–49. `doi:10.1016/j.neucom.2015.03.110`.
URL `https://doi.org/10.1016/j.neucom.2015.03.110`

[17] Y. Lei, X. Chen, M. Min, Y. Xie, A semi-supervised laplacian extreme learning machine and feature fusion with CNN for industrial superheat identification, Neurocomputing 381 (2020) 186–195. `doi:10.1016/j.neucom.2019.11.012`.
URL `https://doi.org/10.1016/j.neucom.2019.11.012`

[18] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, K. McGuinness, Pseudo-labeling and confirmation bias in deep semi-supervised learning, CoRR abs/1908.02983. `arXiv:1908.02983`.
URL `http://arxiv.org/abs/1908.02983`

[19] J. Li, C. Xiong, S. C. H. Hoi, Comatch: Semi-supervised learning with contrastive graph regularization, CoRR abs/2011.11183. `arXiv:2011.11183`.
URL `https://arxiv.org/abs/2011.11183`

[20] Z. Ren, R. A. Yeh, A. G. Schwing, Not all unlabeled data are equal: Learning to weight data in semi-supervised learning, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/f7ac67a9aa8d255282de7d11391e1b69-Abstract.html`

[21] H. Zhang, M. Cissé, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018. URL `https://openreview.net/forum?id=r1Ddp1-Rb`

[22] J. Chen, V. Shah, A. Kyrillidis, Negative sampling in semi-supervised learning, ICML.

[23] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, C. Raffel, Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring, arXiv preprint arXiv:1911.09785.

[24] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. Raffel, E. D. Cubuk, A. Kurakin, C. Li, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/06964dce9addb1c5cb5d6e3d9838f733-Abstract.html`

[25] J. E. van Engelen, H. H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2) (2020) 373–440. `doi:10.1007/s10994-019-05855-6`. URL `https://doi.org/10.1007/s10994-019-05855-6`

[26] D. P. Kingma, M. Welling, Auto-encoding variational bayes, in: Y. Bengio, Y. LeCun (Eds.), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.
URL http://arxiv.org/abs/1312.6114

[27] Z. Wu, Y. Xiong, S. X. Yu, D. Lin, Unsupervised feature learning via non-parametric instance discrimination, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, IEEE Computer Society, 2018, pp. 3733–3742. doi:10.1109/CVPR.2018.00393.
URL http://openaccess.thecvf.com/content_cvpr_2018/html/Wu_Unsupervised_Feature_Learning_CVPR_2018_paper.html

[28] K. He, H. Fan, Y. Wu, S. Xie, R. B. Girshick, Momentum contrast for unsupervised visual representation learning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, IEEE, 2020, pp. 9726–9735. doi:10.1109/CVPR42600.2020.00975.
URL https://doi.org/10.1109/CVPR42600.2020.00975

[29] G. Huang, S. Song, J. N. D. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (12) (2014) 2405–2417. doi:10.1109/TCYB.2014.2307349.
URL https://doi.org/10.1109/TCYB.2014.2307349

[30] W. Lv, Y. Kang, W. X. Zheng, Y. Wu, Z. Li, Feature-temporal semi-supervised extreme learning machine for robotic terrain classification, IEEE Transactions on Circuits and Systems II: Express Briefs (2020) 1–1.

[31] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, T. Raiko, Semi-supervised learning with ladder networks, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information

Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, 2015, pp. 3546–3554.
URL `http://papers.nips.cc/paper/5947-semi-supervised-learning-with-ladder-networks`

[32] L. Guo, T. Han, Y. Li, Robust semi-supervised representation learning for graph-structured data, in: Q. Yang, Z. Zhou, Z. Gong, M. Zhang, S. Huang (Eds.), Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part III, Vol. 11441 of Lecture Notes in Computer Science, Springer, 2019, pp. 131–143. `doi:10.1007/978-3-030-16142-2\_11`.
URL `https://doi.org/10.1007/978-3-030-16142-2_11`

[33] Y. Bengio, A. C. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828. `doi:10.1109/TPAMI.2013.50`.
URL `https://doi.org/10.1109/TPAMI.2013.50`

[34] Y. Zeng, X. Xu, D. Shen, Y. Fang, Z. Xiao, Traffic sign recognition using kernel extreme learning machines with deep perceptual features, IEEE Transactions on Intelligent Transportation Systems 18 (6) (2017) 1647–1653.

[35] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, Y. Kalantidis, Decoupling representation and classifier for long-tailed recognition, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.
URL `https://openreview.net/forum?id=r1gRTCVFvB`

[36] Y. Qing, Y. Zeng, Y. Li, G.-B. Huang, Deep and wide feature based extreme learning machine for image classification, Neurocomputing 412 (2020) 426 – 436.

[37] J. Tang, C. Deng, G. Huang, B. Zhao, Compressed-domain ship detection

on spaceborne optical image using deep neural network and extreme learning machine, IEEE Trans. Geosci. Remote. Sens. 53 (3) (2015) 1174–1185. doi:10.1109/TGRS.2014.2335751.
URL https://doi.org/10.1109/TGRS.2014.2335751

[38] G. Huang, Q. Zhu, C. K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (1-3) (2006) 489–501. doi:10.1016/j.neucom.2005.12.126.
URL https://doi.org/10.1016/j.neucom.2005.12.126

[39] W. Deng, Q. Zheng, L. Chen, Regularized extreme learning machine, in: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009, part of the IEEE Symposium Series on Computational Intelligence 2009, Nashville, TN, USA, March 30, 2009 - April 2, 2009, IEEE, 2009, pp. 389–395. doi:10.1109/CIDM.2009.4938676.
URL https://doi.org/10.1109/CIDM.2009.4938676

[40] X. Jia, R. Wang, J. Liu, D. M. W. Powers, A semi-supervised online sequential extreme learning machine method, Neurocomputing 174 (2016) 168–178. doi:10.1016/j.neucom.2015.04.102.
URL https://doi.org/10.1016/j.neucom.2015.04.102

[41] D. Zabala-Blanco, M. Mora, R. Hernández-García, R. J. Barrientos, The extreme learning machine algorithm for classifying fingerprints, in: 39th International Conference of the Chilean Computer Science Society, SCCC 2020, Coquimbo, Chile, November 16-20, 2020, IEEE, 2020, pp. 1–8. doi:10.1109/SCCC51225.2020.9281232.
URL https://doi.org/10.1109/SCCC51225.2020.9281232

[42] A. Samat, P. Du, S. Liu, J. Li, L. Cheng, E^2lms : Ensemble extreme learning machines for hyperspectral image classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 7 (4) (2014) 1060–1069. doi:10.1109/JSTARS.2014.2301775.
URL https://doi.org/10.1109/JSTARS.2014.2301775

[43] K. Demertzis, L. S. Iliadis, Bio-inspired hybrid intelligent method for detecting android malware, in: S. Kunifuji, G. A. Papadopoulos, A. M. J. Skulimowski, J. Kacprzyk (Eds.), Knowledge, Information and Creativity Support Systems - Selected Papers from KICSS'2014 - 9th International Conference, held in Limassol, Cyprus, on November 6-8, 2014, Vol. 416 of Advances in Intelligent Systems and Computing, Springer, 2014, pp. 289–304. doi:10.1007/978-3-319-27478-2_20.
URL https://doi.org/10.1007/978-3-319-27478-2_20

[44] Y. Jin, Z. Cheng, Z. Chen, C. Chen, X. Jin, B. Sun, A sensorless adaptive optics control system for microscopy based on extreme learning machine, in: 2020 IEEE 6th International Conference on Control Science and Systems Engineering (ICCSSE), 2020, pp. 195–200. doi:10.1109/ICCSSE50399.2020.9171977.

[45] C. Yuan, L. Yang, Robust twin extreme learning machines with correntropy-based metric, Knowl. Based Syst. 214 (2021) 106707. doi:10.1016/j.knosys.2020.106707.
URL https://doi.org/10.1016/j.knosys.2020.106707

[46] S. Zhou, X. Liu, Q. Liu, S. Wang, C. Zhu, J. Yin, Random fourier extreme learning machine with $_{2,1}$-norm regularization, Neurocomputing 174 (2016) 143–153. doi:10.1016/j.neucom.2015.03.113.
URL https://doi.org/10.1016/j.neucom.2015.03.113

[47] Y. Peng, W. Kong, B. Yang, Orthogonal extreme learning machine for image classification, Neurocomputing 266 (2017) 458–464. doi:10.1016/j.neucom.2017.05.058.
URL https://doi.org/10.1016/j.neucom.2017.05.058

[48] K. Zhang, M. Luo, Outlier-robust extreme learning machine for regression problems, Neurocomputing 151 (2015) 1519–1527. doi:10.1016/j.neucom.2014.09.022.
URL https://doi.org/10.1016/j.neucom.2014.09.022

[49] J. Yang, J. Cao, T. Wang, A. Xue, B. Chen, Regularized correntropy criterion based semi-supervised ELM, Neural Networks 122 (2020) 117–129. doi:10.1016/j.neunet.2019.09.030.
URL https://doi.org/10.1016/j.neunet.2019.09.030

[50] H. Pei, K. Wang, Q. Lin, P. Zhong, Robust semi-supervised extreme learning machine, Knowl. Based Syst. 159 (2018) 203–220. doi:10.1016/j.knosys.2018.06.029.
URL https://doi.org/10.1016/j.knosys.2018.06.029

[51] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, Vol. 1, MIT press Cambridge, 2016.

[52] O. Chapelle, J. Weston, L. Bottou, V. Vapnik, Vicinal risk minimization, in: T. K. Leen, T. G. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA, MIT Press, 2000, pp. 416–422.
URL http://papers.nips.cc/paper/1876-vicinal-risk-minimization

[53] O. Chapelle, B. Schölkopf, A. Zien (Eds.), Semi-Supervised Learning, The MIT Press, 2006. doi:10.7551/mitpress/9780262033589.001.0001.
URL https://doi.org/10.7551/mitpress/9780262033589.001.0001

[54] B. Athiwaratkun, M. Finzi, P. Izmailov, A. G. Wilson, There are many consistent explanations of unlabeled data: Why you should average, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019.
URL https://openreview.net/forum?id=rkgKBhA5Y7

[55] N. Liang, G. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Transactions on Neural Networks 17 (6) (2006) 1411–1423.

[56] P. Horata, S. Chiewchanwattana, K. Sunat, Robust extreme learning machine, Neurocomputing 102 (2013) 31–44. doi:10.1016/j.neucom.2011.12.045.
URL https://doi.org/10.1016/j.neucom.2011.12.045

[57] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV, Vol. 9908 of Lecture Notes in Computer Science, Springer, 2016, pp. 630–645. doi:10.1007/978-3-319-46493-0_38.
URL https://doi.org/10.1007/978-3-319-46493-0_38

[58] D. Tanaka, D. Ikami, T. Yamasaki, K. Aizawa, Joint optimization framework for learning with noisy labels, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, IEEE Computer Society, 2018, pp. 5552–5560. doi:10.1109/CVPR.2018.00582.
URL http://openaccess.thecvf.com/content_cvpr_2018/html/Tanaka_Joint_Optimization_Framework_CVPR_2018_paper.html

[59] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: H. Uszkoreit (Ed.), 33rd Annual Meeting of the Association for Computational Linguistics, 26-30 June 1995, MIT, Cambridge, Massachusetts, USA, Proceedings, Morgan Kaufmann Publishers / ACL, 1995, pp. 189–196. doi:10.3115/981658.981684.
URL https://www.aclweb.org/anthology/P95-1026/

[60] P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz, Pruning convolutional neural networks for resource efficient inference, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.
URL https://openreview.net/forum?id=SJGCiw5gl

[61] B. Hariharan, P. A. Arbeláez, R. B. Girshick, J. Malik, Hypercolumns for object segmentation and fine-grained localization, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, IEEE Computer Society, 2015, pp. 447–456. `doi:10.1109/CVPR.2015.7298642`.
URL `https://doi.org/10.1109/CVPR.2015.7298642`

[62] M. Lin, Q. Chen, S. Yan, Network in network, in: Y. Bengio, Y. LeCun (Eds.), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.
URL `http://arxiv.org/abs/1312.4400`

[63] L. Beyer, X. Zhai, A. Oliver, A. Kolesnikov, S4L: self-supervised semi-supervised learning, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, IEEE, 2019, pp. 1476–1485. `doi:10.1109/ICCV.2019.00156`.
URL `https://doi.org/10.1109/ICCV.2019.00156`

[64] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images.

[65] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Ng, Reading digits in natural images with unsupervised feature learning, 2011.

[66] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, I. J. Goodfellow, Realistic evaluation of deep semi-supervised learning algorithms, in: S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada, 2018, pp. 3239–3250.
URL `http://papers.nips.cc/paper/7585-realistic-evaluation-of-deep-semi-supervised-learning-algorithms`

[67] D. T. Nguyen, C. K. Mummadi, T. Ngo, T. H. P. Nguyen, L. Beggel, T. Brox, SELF: learning to filter noisy labels with self-ensembling, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.
URL https://openreview.net/forum?id=HkgsPhNYPS

[68] C. R. Qi, L. Yi, H. Su, L. J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5099–5108.
URL https://proceedings.neurips.cc/paper/2017/hash/d8bf84be3800d12f74d8b05e9b89836f-Abstract.html

Appendices

A. Traditional ELMs

Extreme learning machine (ELM) is an effective learning framework using single-layer feedforward neural networks proposed by Huang [18, 10, 19], which can be used as a classifier. The traditional ELMs [38, 39] consists of two basic characteristics, namely the un-tuned hidden layer and the analytically determined output weights. Concretely, let us assume there are m hidden nodes and the output function of j-th hidden node can be expressed as $g(w_j^T x_i + b_j)$ for sample x_i, where $g(\cdot)$ is activation function and $w_j \in \mathbb{R}^d$, $b_j \in \mathbb{R}$ are the parameters of the hidden nodes randomly assigned based on a certain distribution. Then, based on the output of random hidden layer, the output weight matrix, $\beta = \left[\beta_1, \cdots, \beta_c\right] \in \mathbb{R}^{m\times c}$, is analytically determined by minimizing the least square loss. Specifically, for n sample $(x_i, t_i) \in \mathbb{R}^d \times \mathbb{R}^c$, the objective function of ELM can be presented in a matrix form as $H\beta = T$, where $T = (t_1, \cdots, t_n)^T$ is the label matrix and

$$H = \begin{bmatrix} g\big(w_1^T x_1 + b_1\big) & \cdots & g\big(w_m^T x_1 + b_m\big) \\ \vdots & \ddots & \vdots \\ g\big(w_1^T x_N + b_1\big) & \cdots & g\big(w_m^T x_N + b_m\big) \end{bmatrix} \tag{A.1}$$

A.1. Basic ELM

Note that from then on, the above equation is abbreviated as $H = g(X)$. The least squares optimization problem with ℓ_2-norm can be formalized as

$$\min_{\beta} = \frac{1}{N}\big\|H\beta - T\big\|_F^2 + \gamma\big\|\beta\big\|_F^2 \tag{A.2}$$

Here γ is the penalty term and the solution of the problem can be easily obtained:

$$\begin{aligned} \hat{\beta} &= \left(H^T H + c\boldsymbol{I}\right)^{-1} H^T T \quad if\ n \geq d, \\ \hat{\beta} &= H^T \left(H H^T + c\boldsymbol{I}\right)^{-1} T \quad other. \end{aligned} \tag{A.3}$$

A.2. Online sequential-ELM

The online sequential-ELM (OS-ELM) [55] provides a promising way to process sequential data. It is mainly divided into two phases: initialization and iteration. In the initial phase, the output weight β_0 of the single hidden layer feedforward neural network is obtained through a small number of samples. Suppose there are N_0 samples $\{x_i, t_i\}_{i=1}^{N_0}$, $N_0 > m$, According to Eq.(A.3), we can get

$$\beta_0 = K_0^{-1} H_0^T T_0 \tag{A.4}$$

where, $K_0 = H_0^T H_0$. In the iteration phase, samples are sequentially input to the ELM and the update formula of the output weight matrix β is

$$\begin{aligned} P_{k+1} &= P_k H_{k+1}^T \big(\boldsymbol{I} + H_{k+1} P_k H_{k+1}^T\big) H_{k+1} P_k \\ \beta^{k+1} &= \beta^k + P_{k+1} H_{k+1}^T \big(Y_{k+1} - H_{k+1} \beta^k\big) \end{aligned} \tag{A.5}$$

where $P_{k+1} = K_k^T - K_k^{-1} H_{k+1}^T \big(\boldsymbol{I} + H_{k+1} K_k^{-1} H_{k+1}^T\big) H_{k+1} K_k^{-1}$, H_{k+1} and T_{k+1} are the new data matrix and label matrix respectively.

B. Analysis of SLI-OELM

This section provides detailed proofs related to the SLI-OELM method proposed in this paper.

Firstly, Let us suppose that there are n images and the data matrix is $X_i = \{x_i\}_{i=1}^n$ with noisy one-hot labels matrix $Y_i = \{y_i\}_{i=1}^n$. We shuffle X_i and Y_i

and get reordered X_j and Y_j. The formula is as follows.

$$\begin{aligned}\tilde{X} &= \Lambda X_i + (\boldsymbol{I} - \Lambda) X_j \\ \tilde{Y} &= \Lambda Y_i + (\boldsymbol{I} - \Lambda) Y_j\end{aligned} \tag{B.1}$$

where $\Lambda \in \mathbb{R}^{n \times n}$ is weight diagonal matrix randomly sampled from beta distribution $Be(\alpha, \beta)$ with $\alpha = \beta$. And then the interpolated data matrix $\tilde{X}$ is input to $f_n(\cdot)$ and $g(\cdot)$ in turn, and the hidden layer output matrix $\tilde{H} \in \mathbb{R}^{n \times m}$ is obtained. Then, we formulate the objective function of SLI-ELM as

$$\min_{\beta} \left\| \Lambda^{\frac{1}{2}} (\tilde{H}\beta - Y_i) \right\|_F^2 + \left\| (\boldsymbol{I} - \Lambda)^{\frac{1}{2}} (\tilde{H}\beta - Y_j) \right\|_F^2 + c \left\| \beta \right\|_F^2 \tag{B.2}$$

where Λ is the weight diagonal matrix, $\tilde{H} = g\big(f_n\big(\tilde{X}\big)\big)$, F is Frobenius norm and c represents the coefficient of F-norm.

The analytical solution and iterative solution of Eq.(B.2) are inconvenient to obtain, so we give its alternative form:

$$\min_{\beta} \left\| \tilde{H}\beta - \tilde{Y} \right\|_F^2 + c \left\| \beta \right\|_F^2 \tag{B.3}$$

where $\tilde{Y} = \Lambda Y_i + \big(\boldsymbol{I} - \Lambda\big) Y_j$.

We now show that analytical solution of Eq.(B.2) is equivalent to that of Eq.(B.3).

Proof 1. *The solutions of Eq.(B.2) and Eq.(B.3) are equivalent.*

$$\begin{aligned}&Eq.(B.2) \Longleftrightarrow \min_{\beta} Tr\big(\beta^T \tilde{H}^T \Lambda \tilde{H} \beta + Y_i^T \Lambda Y_i - 2\beta^T \tilde{H}^T \Lambda Y_i\big) + \\ &Tr\big(\beta^T \tilde{H}^T (\boldsymbol{I} - \Lambda) \tilde{H}\beta + Y_j^T (\boldsymbol{I} - \Lambda) Y_j - 2\beta^T \tilde{H}^T (\boldsymbol{I} - \Lambda) Y_j\big) + cTr\big(\beta^T \beta\big) \\ &\Longleftrightarrow \min_{\beta} Tr\big(\beta^T \big(\tilde{H}^T \tilde{H} + c\boldsymbol{I}\big)\beta\big) - 2Tr\big(\beta^T \tilde{H}^T \big(\Lambda Y_i + (\boldsymbol{I} - \Lambda) Y_j\big)\big) + \\ &\underbrace{Tr\big(Y_i^T \Lambda Y_i\big) + Tr\big(Y_j^T (\boldsymbol{I} - \Lambda) Y_j\big)}_{const}\end{aligned} \tag{B.4}$$

The above equation is solved by setting the derivative of Eq.(B.4) to 0. Since the last term has nothing to do with the parameter β, it can be considered as a *constant. So the first derivative of Eq.(B.3) is equivalent to the first derivative of Eq.(B.4), as shown below.*

$$\begin{aligned}\frac{\partial\big(Eq.(B.3)\big)}{\partial\beta} &= \frac{\partial\Big(\big\|\tilde{H}\beta-\big(\Lambda Y_i+(\boldsymbol{I}-\Lambda)Y_j\big)\big\|_F^2+c\big\|\beta\big\|_F^2\Big)}{\partial\beta}\\ &= \frac{\partial\big(Eq.(B.2)\big)}{\partial\beta}\end{aligned}\tag{B.5}$$

C. Analysis of CR-OELM

This section presents the derivation process of the closed-form solution and iterative solution of CR-OELM.

Firstly, we assume that $E(\cdot)$ is a perturbation function representing the small amount δ, such as random rotation, affine or cropping, etc. And there are n images and the data matrix is $X=\{x_i\}_{i=1}^n$ with noisy one-hot labels matrix $Y_i=\{y_i\}_{i=1}^n$ and it's perturbed data matrix is $\acute{X}=E(X)$. Their corresponding hidden layer output matrix are $H\in\mathbb{R}^{n\times m}$ and $\acute{H}\in\mathbb{R}^{n\times m}$ respectively, processed by $g(f(\cdot)_n)$.

The objective function of CR-ELM is as follows

$$\min_{\beta}\|H\beta-Y\|_F^2+c_0\|\beta\|_F^2+c_1\|H\beta-\acute{H}\beta\|_F^2\tag{C.1}$$

where, $c_1\|H\beta-\acute{H}\beta\|_F^2$ is the consistency regularization term, c_1 is penalty coefficient of consistency regularization term.

C.1. The analytical solution of CR-ELM

The derivation process of the analytical solution of Eq.(C.1) is as follows.

$$
\begin{aligned}
Eq.C.1 \Longleftrightarrow \min_{\beta} & Tr\big(\big(H\beta - Y\big)^{T}\big(H\beta - Y\big)\big) + c_0 Tr\big(\beta^{T}\beta\big) \\
& + c_1 Tr\big(\big(H\beta - \acute{H}\beta\big)^{T}\big(H\beta - \acute{H}\beta\big)\big) \\
\Longleftrightarrow \min_{\beta} & Tr\big(\beta^{T}H^{T}H\beta - \beta^{T}H^{T}Y - Y^{T}H\beta + Y^{T}Y\big) + c_0 Tr\big(\beta^{T}\beta\big) \\
& + c_1 Tr\big(\beta^{T}\big(H - \acute{H}\big)^{T}\big(H - \acute{H}\big)\beta\big) \\
\Longleftrightarrow \min_{\beta} & Tr\big(\beta^{T}\big(H^{T}H + c_0\boldsymbol{I} + c_1\big(H - \acute{H}\big)^{T}\big(H - \acute{H}\big)\big)\beta\big) + 2Tr\big(\beta^{T}H^{T}Y\big) \\
& + Tr\big(Y^{T}Y\big) \qquad \text{(C.2)}
\end{aligned}
$$

Then, the first derivative is set to zero:

$$
\frac{\partial Eq.C.2}{\partial \beta} = 2\big(\big(1 + c_1\big)H^{T}H + c_1\big(\acute{H}^{T}\acute{H} - 2H^{T}\acute{H}\big) + c_0\boldsymbol{I}\big)\beta - 2H^{T}Y = 0 \quad \text{(C.3)}
$$

Finally, we get the analytical solution formula (Eq.(C.4)):

$$
\begin{aligned}
\beta^{*} &= \big(\big(1 + c_1\big)H^{T}H + c_1\big(\acute{H}^{T}\acute{H} - 2H^{T}\acute{H}\big) + c_0\boldsymbol{I}\big)^{-1}H^{T}T \quad if\ n \geq d, \\
\beta^{*} &= H^{T}\big(\big(1 + c_1\big)HH^{T} + c_1\big(\acute{H}\acute{H}^{T} - 2H\acute{H}^{T}\big) + c_0\boldsymbol{I}\big)^{-1}T \quad other.
\end{aligned} \qquad \text{(C.4)}
$$

C.2. The iteration form of CR-OELM

Suppose, for any epoch, the k-th batch of samples is defined as $\{X, Y\}_k$. Their random feature matrix and perturbed random feature matrix are $H_k \in \mathbb{R}^{m*p} = g\big(f_n\big(X_k\big)\big)$ and $\acute{H}_k \in \mathbb{R}^{m*p} = g\big(f_n\big(E\big(X_k\big)\big)\big)$ respectively.

At first, we assume that the random features matrix and the perturbed random feature matrix of the 0-th batch samples are H_0, $\acute{H}_0$ respectively. According to Eq.(C.4), the initial parameters of ELM obtained under the 0-th batch samples are:

$$
\begin{aligned}
K_0 &= \big(\big(1 + c_1\big)H_0^{T}H_0 + c_1\big(\acute{H}_0^{T}\acute{H}_0 - 2H_0^{T}\acute{H}_0\big) + c_0\boldsymbol{I}\big) \\
&= H_0^{T}\big(\big(1 + c_1\big)H_0 - 2c_1\acute{H}_0\big) + c_1\acute{H}_0^{T}\acute{H}_0 + c_0\boldsymbol{I} \\
\beta_0 &= K_0^{-1}H_0^{T}Y_0
\end{aligned} \qquad \text{(C.5)}
$$

Then, adding the 1-th batch of samples H_1, $\acute{H}_1$, we perform induction and obtain the iterative relationship of the parameters in 0-th and 1-th batch samples:

$$\beta_1 = \beta_0 + K_1^{-1}\big(H_1^T T_1 - \big(H_1^T\big((1+c_1)H_1 - 2c_1\acute{H}_1\big) + c_1\acute{H}_1^T\acute{H}_1\big)\beta_0\big) \tag{C.6}$$

The derivation process of the iterative relationship between K_1 and K_0 is as follow:

$$\begin{aligned}
K_1 &= \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \\
&\quad + c_1\left(\begin{bmatrix} H_0 \\ H_1 \end{bmatrix} - \begin{bmatrix} \acute{H}_0 \\ \acute{H}_1 \end{bmatrix}\right)^T \left(\begin{bmatrix} H_0 \\ H_1 \end{bmatrix} - \begin{bmatrix} \acute{H}_0 \\ \acute{H}_1 \end{bmatrix}\right) + c_0\boldsymbol{I} \\
&= K_0 + H_1^T H_1 + c_1\big(H_1 - \acute{H}_1\big)^T\big(H_1 - \acute{H}_1\big) \\
&= K_0 + \big(1+c_1\big)H_1^T H_1 + c_1\big(\acute{H}_1^T\acute{H}_1 - 2H_1^T\acute{H}_1\big) \\
&= K_0 + H_1^T\big((1+c_1)H_1 - 2c_1\acute{H}_1\big) + c_1\acute{H}_1^T\acute{H}_1
\end{aligned} \tag{C.7}$$

Thus, the final iterative formula for K_{k+1}, K_k, β_{k+1} and β_k can be induced as shown in Eq.(C.8).

$$\begin{aligned}
K_{k+1} &= K_k + H_{k+1}^T\big((1+c_1)H_{k+1} - 2c_1\acute{H}_{k+1}\big) + c_1\acute{H}_{k+1}^T\acute{H}_{k+1} \\
\beta_{k+1} &= \beta_k + K_{k+1}^{-1}\big(H_{k+1}^T Y_{k+1} - \big(H_{k+1}^T\big((1+c_1)H_{k+1} - 2c_1\acute{H}_{k+1}\big) \\
&\qquad + c_1\acute{H}_{k+1}^T\acute{H}_{k+1}\big)\beta_k\big)
\end{aligned} \tag{C.8}$$