Compensation Learning

Anonymous

Abstract

Weighting strategy prevails in machine learning. For example, a common approach in robust machine learning is to exert lower weights on samples which are likely to be noisy or hard. This study reveals another undiscovered strategy, namely, compensating, that has also been widely used in machine learning. Learning with compensating is called compensation learning and a systematic taxonomy is constructed for it in this study. In our taxonomy, compensation learning is divided on the basis of the compensation targets, inference manners, and granularity levels. Many existing learning algorithms including some classical ones can be seen as a special case of compensation learning or partially leveraging compensating. Furthermore, a family of new learning algorithms can be obtained by plugging the compensation learning into existing learning algorithms. Specifically, three concrete new learning algorithms are proposed for robust machine learning. Extensive experiments on text sentiment analysis, image classification, and graph classification verify the effectiveness of the three new algorithms. Compensation learning can also be used in various learning scenarios, such as imbalance learning, clustering, regression, and so on.

1. Introduction

In supervised learning, a loss function is defined on the training set, and the training goal is to seek optimal models by minimizing the training loss. According to the degree of training difficulty, samples can be divided into easy, medium, hard, and noisy samples. Particularly, hard samples are defined as those that are beyond the learning capability of the utilized learning models in this study. Generally, easy and medium samples are indispensable and positively influence the training. The whole training procedure can significantly benefit from medium samples if appropriate learning manners are leveraged. However, the whole training procedure is vulnerable to noisy and hard samples.

A common practice is to introduce the weighting strategy if hard and noisy samples exist. Low weights are assigned to noisy and hard samples to reduce their negative influences in loss minimization. This strategy usually infers the weights and subsequently conducts training on the basis of the weighted loss. Liu and Tao [26] investigated the classification when labels are randomly corrupted. They defined a conditional distribution-based weighting strategy to deal with noisy labels. Wang et al. [35] proposed a Bayesian method to infer the sample weights as latent variables. Kumar et al. [23] proposed a self-paced learning manner that combines the two steps as a whole by using an added regularizer. Meta learning [24, 31, 38] is introduced to alternately infer weights and seek model parameters with an additional validation set.

Various robust learning methods exist that do not rely on the weighting strategy. For example, the classical method support vector machine (SVM) [4] introduces slack variables to address possibly hard/noisy samples, and robust clustering [6] introduces additional vectors to cope with noises. However, a unified theory to better explain such methods and subsequently illuminate more novel methods remains lacking. In this study, another under-explored yet widely used strategy, namely, compensating, is revealed and further investigated. Many existing learning methods including some classical ones can be considered introducing or partial on the basis of compensating. Learning with compensating is called compensation learning in this study.

We conduct a pilot study for compensation learning in terms of theoretical taxonomy, connections with existing classical learning methods, and new concrete compensation learning algorithms. First, four compensation targets, three inference manners, and three compensation granularities are defined. Second, four existing learning methods are re-explained in the viewpoint of compensation learning. Third, three concrete compensation learning algorithms are proposed, namely, logit compensation with $l1$-regularization, label compensation with $l1$-regularization, and mixed label compensation. Last, the three proposed learning algorithms are evaluated on data corpora from text sentiment classification, image classification, and graph classification.

Our main contributions are summarized as follows:

1) An under-explored yet widely used learning strategy, namely, compensating, is discovered and formalized in this study. A new learning paradigm, named compensation learning, is presented and a taxonomy is constructed for it. In addition to the robust learning mainly referred in this paper, other learning scenarios, such as imbalance learning

can also benefit from compensation learning.

2) Four typical existing learning methods are discussed and re-explained with the viewpoint of compensation learning. New insights can be obtained when existing methods are placed in the compensation learning framework. Theoretically, many new methods can be generated on the basis of the introducing the idea of compensating into existing methods.

3) Three concrete new compensation learning methods are proposed. Experiments on robust learning on eight data sets verify their effectiveness compared with two classical methods including soft and hard Bootstrapping losses.

2. Related work

2.1. Weighting strategy in machine learning

Weighting is a widely used machine learning strategy in at least the following five areas, including, noisy-aware learning [29], curriculum learning [1], crowdsourcing learning [5], cost-sensitive learning [3], and imbalance learning [18]. In noisy-aware and curriculum learning areas, weights are sample-wise; in cost-sensitive learning, weights can be either sample-wise, category-wise, or mixed; in imbalance learning, weights are category-wise; in crowdsourcing learning, weights can be either sample-wise, label-wise, or subset-wise.

In the sample-wise context, samples are divided into easy-level, medium-level, hard-level, and noisy. Intuitively, the weights of the last two types of samples should be reduced; the weights of easy samples are kept or reduced; and the weights of medium samples are kept or enlarged. For example, in Focal loss [25], the weights of easy samples are (relatively) reduced and those of the medium[1] samples are (relatively) enlarged. Most existing studies do not assume the above division. Instead, samples are usually divided into easy/non-easy or normal/noisy. For example, in Focal loss and AdaBoost [7], the weights of non-easy samples are gradually increased.

In cost-sensitive learning, the weights are associated with the pre-determined costs. In imbalance learning, categories with lower proportions are usually negatively affected. Therefore, increasing the weights of samples in the low-proportion categories is a common practice.

The compensating strategy investigated in this study does not intend to eliminate the weighting strategy. Instead, this study summarizes various existing learning ideas which do not utilize weighting yet. These learning ideas are systematically investigated to attribute to a new learning paradigm, namely, compensation learning. These two strategies can be mutually enhanced[2]. Theoretically, each concrete weighting-based learning method may correspond to a concrete compensating-based learning method. A solid and deep investigation for the weighting strategy in machine learning will significantly benefit compensation learning.

2.2. Noise-aware machine learning

This study investigates compensation learning mainly in robust machine learning, or exactly, noise-aware learning. The weighting strategy is prevailing in this area. There exist two common technical solutions.

In the first solution, noise detection is performed and noisy samples may assign lower weights in the successive model training. Koh and Liang [12] defined an influence function to measure the impact of each sample on the model training. Samples with higher influence scores are more likely to be noisy. Huang et al. [19] conducted a cyclical pre-training strategy and recoded the training losses for each sample in the whole cycles. The samples with higher average training losses are more likely to be noisy.

In the second solution, an end-to-end procedure is leveraged to construct a noisy-robust model. Reed et al. [30] proposed a Bootstrapping loss to reduce the negative impact of samples which may be noisy. Goldberger and en-Reuven [9] designed a noise adaptation layer to model the relationship between overserved labels that may be noisy and true latent labels.

A recent survey can be referred to [11]. Compensation learning can replace weighting in both above solutions. In this study, only the second solution is referred.

3. Taxonomy of compensation learning

Compensating can be used in many learning scenarios. This section leverages classification as the illustrative example. Given a training set $S = \{x_i, y_i\}$, $i = 1, \cdots, N$, where x_i is the i-th sample, and y_i is its categorical label. In a standard supervised deep learning context, let u_i be the logit vector for x_i output using a deep neural network. The training loss can be written as follows:

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{N} l(\text{softmax}(DNN(x_i)), y_i) \\ &= \sum_{i=1}^{N} l(\text{softmax}(u_i), y_i), \end{aligned} \tag{1}$$

where softmax() transforms the logit vector u_i into a soft label p_i and *DNN* represents a deep neural network.

In the weighting strategy, the loss function is usually defined as follows:

[1] Although Focal loss call these samples as 'hard', we place them in the 'medium' category. The 'hard' samples in this paper are defined as those beyond the learning capability of the involved learning model.

[2] For example, a sample-level weighting method (*e.g.*, Focal loss) can be transformed into a category-level weighting method (*e.g.*, replace the sample-level prediction y_i with the category-level average y_c) inspired by our taxonomy for compensating learning.

$$\mathcal{L} = \sum_{i=1}^{N} w_i \cdot l(\text{softmax}(u_i), y_i), \tag{2}$$

where w_i is the weight associated to the sample x_i. Theoretically, easy and medium samples are valuable in training. To train an effective model, the weights of medium samples should be enlarged or at least not reduced, whereas the weights of noisy and hard samples should be reduced[3]. The weights are inferred with the two solutions introduced in Section 2.2. The more likely a sample belongs to noisy, the lower its weight.

The compensating strategy investigated in this study can also increase or reduce the influences of samples in model training on the basis of their degrees of training difficulty. For instance, a negative value can be added to reduce the loss incurred from a noisy sample. Resultantly, the negatively influence of this sample will be reduced because its impact on gradients is reduced. Contrarily, when the influence should be increased, a positive value can be added to the loss incurred from the sample. In terms of mathematical computation, "weighting" relies on the multiplication operation, whereas "compensating" relies on adding operation.

To better investigate compensating, learning with compensating is called compensation learning in this study. In the following subsections, the whole compensation learning taxonomy in terms of compensation targets, inference manners, and granularity levels is described.

3.1. Compensation targets

Eq. (3) contains four different types of variables for each sample, namely, raw feature x_i, logit vector u_i, label y_i, and sample loss l_i. Therefore, compensation targets can be feature, logit vector, label, and loss.

Compensation for feature (Feature compensation). In this kind of compensation, the raw feature vector (x_i) or transformed feature vectors (*e.g.*, dense feature) of each sample can have a compensation vector (Δx_i). Eq. (1) becomes

$$\begin{aligned} \mathcal{L} &= \sum\nolimits_i l(\text{softmax}(DNN(x_i+\Delta x_i)), y_i) \\ &= \sum\nolimits_i l(\text{softmax}(u_i'), y_i). \end{aligned} \tag{3}$$

Compensation for logit vector (Logit compensation). In this kind of compensation, the logit vector (u_i) of each sample can have a compensation vector (Δu_i). Eq. (1) becomes

$$\mathcal{L} = \sum\nolimits_i l(\text{softmax}(u_i+\Delta u_i), y_i). \tag{4}$$

Compensation for label (Label compensation). In this kind of compensation, the label (y_i) of each sample can have a compensation label (Δy_i). Let $p_i = \text{softmax}(u_i)$. Eq. (1) becomes

$$\begin{array}{ll} \text{(i)} & \mathcal{L} = \sum_i l(p_i, y_i+\Delta y_i) \quad \text{or} \\ \text{(ii)} & \mathcal{L} = \sum_i l(p_i+\Delta y_i, y_i). \end{array} \tag{5}$$

In Eq. (5-i), Δy_i is added to the true label y_i, while in (ii) Δy_i is added to the predicted label p_i. Considering that labels after compensation should be a (soft) label, Δy_i should satisfy the following requirements:

$$\sum\nolimits_c \Delta y_{ic} = 0 \quad \text{and} \quad y_{ic} + \Delta y_{ic} \geq 0 \quad \text{or} \quad p_{ic} + \Delta y_{ic} \geq 0. \tag{6}$$

Compensation for loss (Loss compensation). In this kind of compensation, the loss of each sample can have a compensation loss (Δl_i). Eq. (1) becomes

$$\mathcal{L} = \sum\nolimits_i l(\text{softmax}(u_i), y_i)+\Delta l_i. \tag{7}$$

Remark: The compensation variables (*i.e.*, Δx_i, Δu_i, Δy_i, and Δl_i) are trainable during training. They are introduced to reduce the negative impact of some training samples (*e.g.*, noisy or hard ones) and increase the positive impact of some samples (*e.g.*, medium ones). For example, in raw feature-based compensation, let m_c be the center vector of the c-th category x_i belongs to. Ideally, if $\Delta x_i = m_c - x_i$, the impact of x_i is completely reduced if x_i is noisy. The compensation variables that can reduce the influence of samples are called downstream compensation, whereas those that can increase the influence are called upstream.

If the loss functions defined in Eqs. (3)–(7) are directly used without any other constrictions on the compensation variables, nothing can be learned as compensation variables are trainable. For example, when the loss in Eq. (4) is directly used, a random model will be produced because in the training, the value of Δu_i will be learned to be equal to y_i. How to infer them and learn with the above loss functions are described in the succeeding subsection.

There may exist other compensation candidates, such as view and gradient, which will be explored in future work.

3.2. Compensation inference

In compensation learning, compensation variables in losses in Eqs. (3)–(7) should be inferred during training. There exist three manners (maybe not exhaustive) to infer their values and optimize the whole loss.

Inference with regularization. In this manner, a regularization term is added for the compensation variables. For example, a natural assumption is that the samples that require the compensation variables occupy a little. Therefore, $l1$-norm can be used. Taking the logit and label compensation as examples. Two loss functions are defined as follows:

$$\begin{array}{l} \text{(i)}\ \mathcal{L} = \sum_i l(\text{softmax}(u_i + \Delta u_i), y_i) + \lambda \text{Reg}(\Delta u_i), \\ \text{(ii)}\ \mathcal{L} = \sum_i l(p_i, y_i + \Delta y_i) + \lambda \text{Reg}(\Delta y_i), \end{array} \tag{8}$$

[3] Although some studies claim that they mine the hard samples and enlarge their weights, in our opinion, the mentioned hard samples should be medium samples or the relatively hard ones in the medium level. In this study, hard samples are those beyond the capability of learning models.

where λ is a hyper-parameter and Reg() is regularizer. This manner is similar to the self-paced learning [23]. When $\lambda \to \infty$, no compensation is allowed and compensation learning is reduced to conventional learning.

Inference with Meta learning. In this manner, the compensation variables are inferred on the basis of another small clean validation set with Meta learning. Given a clean validation set Ω comprising M clean training samples and taking loss compensation as an example. Let κ_i be the loss compensation variable for x_i ($\in S$). We first define that

$$\mathcal{L}(\kappa) = \sum\nolimits_{i \in S} l(\text{softmax}(u_i), y_i : \Theta) + \kappa_i, \tag{9}$$

where Θ is the model parameter set to be learned. Given κ, Θ can be obtained by solving

$$\Theta^*(\kappa) = \arg\min_{\Theta} \sum\nolimits_{i \in S} l(\text{softmax}(u_i), y_i : \Theta) + \kappa_i. \tag{10}$$

After Θ is obtained, κ can be optimized by solving

$$\kappa^* = \arg\min_{\kappa} \sum\nolimits_{j \in \Omega} l(\text{softmax}(u_j), y_j : \Theta^*(\kappa)). \tag{11}$$

These two optimization can be performed alternately, and finally Θ and κ are learned. When either logit or label compensation is used, the above optimization procedure can also be utilized with slight variations.

The above inference manner is similar with that used in the Meta learning-based weighting strategy for robust learning [31]. Meta learning has been widely used in robust learning and many existing Meta learning-based methods [24, 38] can be leveraged for compensation learning.

Inference with prior knowledge. In this manner, the compensation variables are inferred on the basis of prior knowledge. Alternatively, the compensation variables are fixed before the optimizing of training loss. Taking the label compensation as an example. Given that for each sample, we can obtain a predicted label y'_i by another model, the label compensation can be defined as

$$\Delta y_i = \alpha(y'_i - y_i), \tag{12}$$

where α is a hyper-parameter and locates in [0, 1]. Δy_i defined in Eq. (12) satisfies the condition given by Eq. (6). If y'_i is in trust, it is highly possible that Δy_i approaches to zero if x_i is normal and it is large if x_i is noisy. Assuming that y'_i is the output of the model in the previous epochs. Eq. (5) becomes

$$\begin{aligned} \mathcal{L} &= \sum\nolimits_i l(\text{softmax}(u_i), y_i + \alpha(y'_i - y_i)) \\ &= \sum\nolimits_i l(\text{softmax}(u_i), (1-\alpha) y_i + \alpha y'_i), \end{aligned} \tag{13}$$

which is exactly the soft Bootstrapping loss defined in [30].

3.3. Compensation granularity

Compensation granularity has three levels.

Sample-level compensation. All the compensation variables discussed above are for samples. Each sample has its own compensation variable.

Category-level compensation[4]**.** In this level, samples within the same category share the same compensation. Taking the logit vector-based compensation as an example, when category-level compensation is utilized, the loss in Eq. (4) becomes

$$\mathcal{L} = \sum\nolimits_i l(\text{softmax}(u_i + \Delta u_{y_i}), y_i). \tag{14}$$

Category-level compensation mainly solves the problem when the impact of all the samples of a category should be increased. For example, in long-tail learning, the tail category should be emphasized in learning.

Mixed-level compensation[5]**.** In this level, sample- and category- levels are considered. This case occurs in complex contexts, *e.g.*, when noisy labels and category imbalance exist. Taking label-based compensation as an example. The loss in Eq. (4) can be written as

$$\mathcal{L} = \sum\nolimits_i l(p_i, y_i + \Delta y_i + \Delta y_{y_i}), \tag{15}$$

where Δy_{y_i} is the category-level label compensation.

4. Connection with existing learning paradigms

Weighting strategy is straightforward and quite intuitive, hence it has been widely used in the machine learning community. Compensating seems not as straightforward as weighting. However, it can play the same/similar role with weighting in machine learning. They both have their own strengths. Compensating can be used in the feature, logit vector, label, and loss, whereas weighting is usually used in the loss. Weighting can be used inside of a sample (*e.g.*, attention in deep learning), whereas compensation learning may be not. A theoretical comparison for them is beneficial for both strategies and we leave it as future work.

Many classical and newly proposed learning methods can be attributed to compensation learning or explained in the viewpoint of compensation learning. We choose the following four methods as illustrative examples.

SVM [4]. This model is based on the following hinge loss

$$l_i = \max(0, 1 - y_i(\mathrm{w}^T x_i + b)). \tag{16}$$

To reduce the negative contributions of noisy or hard samples, the loss can be compensated as follows

$$\begin{aligned} l'_i &= \max(0, l_i - \xi_i) \\ &= \max(0, 1 - y_i(\mathbf{w}^{\mathrm{T}} x_i + b) - \xi_i) \qquad (\xi_i \geq 0). \end{aligned} \tag{17}$$

[4]In some methods, all the samples share the same compensation.

[5]There can be mixed compensation targets and mixed inference manners.

Then the whole training loss with max margin and $l1$-norm for ξ_i becomes

$$\mathcal{L} = \frac{1}{2}\|\mathbf{w}\|^2 + \sum\nolimits_i l'_i + \lambda|\xi_i| \qquad (\xi_i \geq 0). \tag{18}$$

The minimization of Eq. (18) equals to the following optimization problem:

$$\begin{array}{l} \min\limits_{\mathbf{w},b,\{\xi_i\}} \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_i \xi_i \\ \text{s.t.} \quad 1 - y_i(\mathbf{w}^T x_i + b) - \xi_i \leq 0 \\ \qquad \xi_i \geq 0, i = 1, \cdots, N \end{array}, \tag{19}$$

which is the standard form of SVM (without kernel). Alternatively, the slack variable can be seen as a loss compensation for SVM. Naturally, other types of compensation (*e.g.*, label compensation) may be considered in SVM.

Robust clustering [6]. Let m_c be the cluster center of the c-th cluster. Let u_{ic} ($\in \{0, 1\}$) denote whether x_i belongs to the c-th cluster. The optimization form of conventional data clustering can be written as follows:

$$\min_{\{m_c\},\{u_{ic}\}} \sum_{i=1}^{N} \sum_{c=1}^{C} u_{ic} \|x_i - m_c\|_2^2. \tag{20}$$

Given that outlier samples may exist, sample-level feature compensation (denoted as o_i for x_i) can be introduced with regularization. (20) becomes

$$\min_{\{m_c\},\{u_{ic}\},\{o_i\}} \sum_{i=1}^{N} \sum_{c=1}^{C} u_{ic} \left(\|(x_i + o_i) - m_c\|_2^2 + \lambda \text{Reg}(o_i) \right). \tag{21}$$

When $l2$-norm is used, the optimization of (21) becomes the method proposed by Foreo et al. [6] as follows

$$\min_{\{m_c\},\{u_{ic}\},\{o_i\}} \sum_{i=1}^{N} \sum_{c=1}^{C} u_{ic} \left(\|(x_i + o_i) - m_c\|_2^2 + \lambda \|o_i\|_2 \right). \tag{22}$$

Knowledge distillation [15]. In knowledge distillation, there are two deep neural networks called teacher and student, separately. The output of the teacher model for x_i is as follows:

$$q_i = \text{softmax}(z_i/T), \tag{23}$$

where z_i is the logit vector from the teacher model and T is the temperature. q_i can be viewed as a prior knowledge about the label compensation for the student model. Then according to Eq. (13), the training loss of the student model with label compensation becomes

$$\begin{aligned} \mathcal{L} &= \sum\nolimits_i l(p_i, y_i + \alpha(q_i - y_i)) \\ &= \sum\nolimits_i l(p_i, (1 - \alpha)y_i + \alpha q_i), \end{aligned} \tag{24}$$

which is exactly the loss function of knowledge distillation.

Logit adjustment-based imbalance learning [28]. In a multi-category classification problem, let π_c be the proportion of the training samples in the c-th category. Let $\mathbf{g} = [g(\pi_1), \ldots, g(\pi_C)]$. When the proportions are imbalanced, a category-level of logit compensation can be introduced as follows:

$$\mathcal{L} = \sum\nolimits_i l\left(\text{softmax}(u_i + \mathbf{g}), y_i\right). \tag{25}$$

For the above loss, when $g()$ is an increasing function, we conjecture that the influences of samples in the minority categories (*i.e.*, $\pi_c < \frac{1}{C}$) on the loss are increased. We have not obtained a strict proof for this conjecture. Nevertheless, it is established for two special categories.

Lemma 1: *When $g()$ is an increasing function and cross-entropy is used, the logit compensation-based loss defined in Eq. (25) increases the contributions of samples in the category with the smallest proportion while reduces those of samples in the one with the largest proportion.*

The proof is in the supplementary materials. As the influences of samples in the minority categories on the loss are increased, the imbalanced problem can be alleviated by the logit compensation used in (25). When $g(\pi_c) = \tau \log(\pi_c)$ $(\tau > 0)$ and cross-entropy loss are used, (25) becomes

$$\mathcal{L} = -\sum\nolimits_i \log \frac{e^{u_{i,y_i} + \tau \log \pi_{y_i}}}{\sum_c e^{u_{i,c} + \tau \log \pi_c}}, \tag{26}$$

which is exactly the logit adjusted loss in [28].

Other typical methods such as Robust nonrigid ICP [17], ISDA [36], D2L [27], DAC (both weighting and compensating are used) [33], Deep self-learning [13], LDAM [2], MRFL [40], Robust linear regression [32], and Bootstrapping loss [30] can also be explained with compensation learning. The discussion is omitted due to lack of space.

5. Three new learning methods

This section introduces three method examples by introducing the idea of compensation learning into existing algorithms. These three methods have been partially mentioned in Section 3. The algorithmic steps are omitted due to lack of space and attached in the supplementary materials.

5.1. $L1$-based logit compensation (Logit-$l1$)

An example is given to explain how logit compensation works. Assume that the inferred logit vector of a noisy sample x_i is $u_i = [3.0, 0.8, 0.2]$ and its (noisy) label y_i is [0, 1, 0]. The cross-entropy loss incurred by this training sample is 2.36. This loss negatively affects in training because y_i is noisy. To reduce the negative influence, if a compensation vector (*e.g.*, [-1, 2, 0]) is learned, the new logit vector becomes [2, 2.8, 0.2]. Consequently, the cross-entropy loss of x_i is 0.42, which is much lower than 2.36. When $l1$-norm is used, the training loss is

$$\mathcal{L} = \sum\nolimits_i l(\text{softmax}(u_i + v_i), y_i) + \lambda|v_i|, \tag{27}$$

where v_i is the logit compensation vector and it is trainable during the training stage. If no noisy and hard samples are present, v_i will approach to zero for all training samples. This method is called Logit-$l1$ for brevity.

5.2. $L1$-based label compensation (Label-$l1$)

Similar with Logit-$l1$, an $l1$-based label compensation method is also obtained. The training loss in Eq. (8-i) becomes the following:

$$\begin{aligned} &\rho_i = \mathrm{sigmoid}(\eta_i), \\ &\mathcal{L} = \sum_i l(p_i, y_i + \rho_i(p_i - y_i) + \lambda|\rho_i|, \end{aligned} \tag{28}$$

where sigmoid() is used to ensure that $\rho_i \in [0, 1]$.

For noisy samples, an effective model is more likely to output the true labels instead of noisy labels even in the early stage of training. Therefore, their corresponding ρ values should be close to 1 and the negative influences of noisy samples can thus be reduced on the training loss. This method is called Label-$l1$ for brevity.

5.3. Mixed label compensation (Mixed label)

As described in Eq. (13), the Bootstrapping loss can be viewed as a compensated loss when y'_i is the output of the model trained in the previous epochs. Eq. (13) is a kind of sample-level label compensation. It can be naturally extended into the category-level defined as follows

$$\mathcal{L} = \sum_i l(p_i, y_i + \lambda(p_{y_i} - y_i)), \tag{29}$$

where p_{y_i} is the average prediction from the model trained in the previous epochs for the samples in the y_i category and is calculated as follows

$$p_{y_i} = \sum_{j: y_j = y_i} p_j / N_{y_i}. \tag{30}$$

Moreover, a mixed label compensation method can be obtained with the following loss:

$$\mathcal{L} = \sum_i l(p_i, y_i + \alpha(\beta p_{y_i} + (1-\beta)p_i - y_i)), \tag{31}$$

where α and β are hyper-parameters and are located in [0, 1]. When β equals 0, the above loss becomes the soft Bootstrapping loss. When β equals 0 and p_i is transformed to one-hot, the loss becomes the hard Bootstrapping loss.

6. Experiments

This section evaluates the three proposed compensation learning methods in three tasks including sentiment analysis, image classification, and graph classification.

6.1. Competing methods

As the three proposed methods belong to the end-to-end noise-aware solution, the following methods are compared in the experiments.

Base method. The base methods in the three tasks are BiLSTM with attention [8], ResNet-18 [14], and GCN [21], respectively.

Base method with soft Bootstrapping loss. This method is based on the soft Bootstrapping loss instead of the raw loss. It is called soft Bootstrapping for brevity.

Base method with hard Bootstrapping loss. This method is based on the hard Bootstrapping loss instead of the raw loss. It is called hard Bootstrapping for brevity.

Base method with our Logit-$l1$ loss. This method is based on the loss defined in Eq. (27). It is called Logit-$l1$ for brevity.

Base method with our Label-$l1$ loss. This method is based on the loss defined in Eq. (28). It is called Label-$l1$ for brevity.

Base method with our Mixed label. This method is based on the loss defined in Eq. (31). It contains two sub-versions which are called Mixed soft and Mixed hard (p_i is one-hot) for brevity, respectively.

The parameter settings are different in the three tasks and detailed in the corresponding subsections as well as the supplementary materials. In all experiments, the average classification accuracy of three repeated runs is recorded for each setting. All the source codes are available in the supplementary materials.

6.2. Text sentiment analysis

Three benchmark sentimental analysis data sets are used, namely, IMDB, SST-2, and MR [34]. All are binary tasks. The details of these three data sets can be seen in [34]. Two types of label noises are added to the three data sets. In the first type (symmetric), the labels of the former 0%, 5%, 10%, and 20% training samples are flipped to simulate the label noises; in the second type (asymmetric), the labels of the former 5%, 10%, and 20% positive samples are flipped to negative.

The 300-D Glove [40] embedding is used. The settings for #epochs, batch size, learning rate, and dropout rate follow the settings in [16, 37]. The value of λ in Logit-$l1$ and Label $l1$ is set as 0.75. The data split and other parameter settings are detailed in the supplementary materials.

The results of the competing methods on the IMDB and SST-2 for the symmetric and asymmetric label noises are shown in Tables 1 and 2. The results on MR are presented in the supplementary materials. Our proposed method, Logit-$l1$, achieves the overall best results (15 highest accuracies among 21 comparisons). When no added label noises are present (0%), Logit-$l1$ still achieves slightly better results than the base method BiLSTM with attention. Mixed hard/soft outperform hard/soft Bootstrapping suggesting that the combination of both sample-level and category-level information is useful.

Given that judging the sentimental states of some sentences is difficult, inevitably, some original samples are hard or noisy. When Logit-$l1$ is used, some original labels with higher $l1$-norm values of logit compensation are found to

		Symmetric noise			Asymmetric noise		
	0%	5%	10%	20%	5%	10%	20%
Base (BiLSTM+attention)	84.39%	83.04%	81.90%	78.13%	82.35%	79.53%	73.74%
Soft Bootstrapping	84.79%	83.87%	81.11%	79.60%	83.36%	80.70%	73.52%
Hard Bootstrapping	84.44%	84.10%	83.01%	80.84%	82.48%	81.42%	75.26%
Mixed soft	84.79%	83.87%	82.20%	80.39%	83.36%	80.95%	74.24%
Mixed hard	84.74%	84.10%	83.01%	80.84%	83.00%	**82.43%**	75.26%
Label-$l1$	85.04%	83.61%	83.63%	81.37%	83.77%	81.54%	70.67%
Logit-$l1$	**85.17%**	**84.53%**	**83.75%**	**81.64%**	**84.45%**	81.44%	**76.87%**

Table 1. Classification accuracies on IMDB.

		Symmetric noise			Asymmetric noise		
	0%	5%	10%	20%	5%	10%	20%
Base (BiLSTM+attention)	83.85%	82.71%	81.12%	79.72%	82.07%	81.46%	79.49%
Soft Bootstrapping	83.77%	83.25%	**82.21%**	80.40%	82.78%	81.66%	79.14%
Hard Bootstrapping	83.68%	83.18%	81.45%	80.50%	82.25%	81.73%	79.52%
Mixed soft	84.00%	83.25%	**82.21%**	**80.71%**	82.78%	81.81%	79.14%
Mixed hard	83.89%	**83.31%**	81.69%	80.50%	82.55%	82.01%	**79.82%**
Label-$l1$	83.91%	83.13%	81.76%	80.21%	82.06%	82.08%	79.57%
Logit-$l1$	**84.10%**	83.18%	81.83%	80.42%	**82.80%**	**82.23%**	78.87%

Table 2. Classification accuracies on SST-2.

be error. For example, the sentence "*Plummer steals the show without resorting to camp as nicholas' wounded and wounding uncle ralph*" is labeled as positive in the original set. Some other samples with higher $l1$-norm values of logit compensation are difficult to judge, *e.g.*, "*A different movie – sometimes tedious – by a director many viewers would like to skip but film buffs should get to know.*" More examples are listed in the supplementary materials.

On IMDB, the Base model is usually converged in the second epoch. However, Logit-$l1$ is usually converged in the third or the fifth epoch. The validation accuracies of the five epochs for the Base model and our Logit-$l1$ are shown in Figure. 1. Logit-$l1$ can decelerate the convergence speed leading that the training data can be more fully trained.

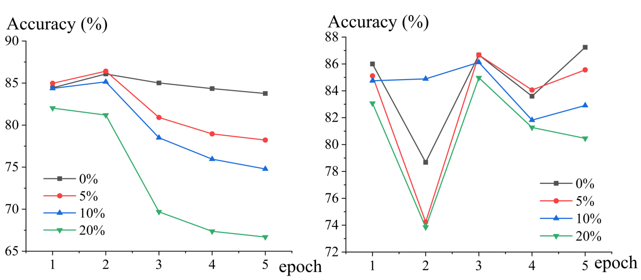

Figure 1. The validation accuracies in the first five epochs under different proportions of random noises on IMDB when using Base (the left figure) and Logit-$l1$ (the right), respectively.

We plot the distribution of $l1$-norm of compensated logit vectors when using Logit-$l1$ on the training set. The results are shown in the supplementary materials. All distribution curves show a long-tail trend, which is quite reasonable.

6.3. Image classification

Two benchmark image classification data sets, namely, CIFAR-10 and CIFAR-100 [22], are used. CIFAR-10 contains 10 categories and CIFAR-100 contains 100. The details of these two data sets are shown in [22].

The synthetic label noises are simulated on the basis of the two common schemes used in [10, 12, 20]. The first is the random scheme in which each training sample is assigned to a uniform random label with a probability p. The second is the pair scheme in which each training sample is assigned to the category next to its true category on the basis of the category list with a probability p. The probability p is set as 0%, 10%, 20%, and 30% in this experiment.

Label-$l1$ is not involved in this experiment given that tuning the learning rate is difficult. The parameter settings are as follows. The batch size, learning rate, and #epochs are set as 128, 0.01, and 150, respectively; the proportion of train/val/test data is 4:1:1 according to the default split. Other parameter settings are detailed in the supplementary materials.

The results are shown in Tables 3 and 4. Both Logit-$l1$ and Mixed hard achieve five highest accuracies among the fourteen comparisons. On CIFAR-10, our methods (Logit-$l1$, Mixed soft, and Mixed hard) are better than others on the random scheme. On the pair scheme, Logit-$l1$ produces the highest accuracies. On CIFAR-100, Mixed hard outperforms others on the random setting when $p < 30\%$. On the pair setting of CIFAR-100, our proposed methods do not produce better results.

		Random noise			Pair noise		
	0%	10%	20%	30%	10%	20%	30%
Base (ResNet-18)	91.96%	86.21%	80.55%	78.53%	87.15%	83.06%	80.86%
Soft Bootstrapping	91.79%	86.26%	80.46%	77.82%	87.14%	83.46%	80.93%
Hard Bootstrapping	91.76%	86.33%	80.44%	77.66%	87.30%	82.90%	80.42%
Mixed soft	**92.19%**	87.06%	80.74%	77.82%	87.21%	83.46%	81.04%
Mixed hard	92.13%	**87.20%**	81.13%	**78.92%**	87.30%	83.19%	81.90%
Logit-$l1$	91.96%	86.78%	**81.35%**	78.65%	**87.95%**	**84.16%**	**82.57%**

Table 3. Classification accuracies on CIFAR-10.

		Random noise			Pair noise		
	0%	10%	20%	30%	10%	20%	30%
Base (ResNet-18)	70.67%	64.82%	58.52%	**52.66%**	**66.38%**	60.39%	52.89%
Soft Bootstrapping	70.54%	64.29%	58.84%	52.33%	66.09%	60.33%	52.45%
Hard Bootstrapping	70.76%	64.77%	58.89%	52.07%	66.17%	60.21%	52.88%
Mixed soft	70.96%	65.05%	59.39%	52.33%	66.33%	60.68%	**53.12%**
Mixed hard	**71.25%**	**65.16%**	**59.58%**	52.33%	66.27%	60.46%	52.88%
Logit-$l1$	70.56%	64.71%	59.12%	52.35%	65.95%	**60.74%**	52.77%

Table 4. Classification accuracies on CIFAR-100.

6.4. Graph classification

Graph classification in this experiment refers to the task of graph node classification. The inductive mode is used in which a training graph with all labels is fed into a model, such as GCN [21]. The trained model is then used to predict the categorical labels of all nodes in a test graph.

Three benchmark graph classification data sets, namely, Citeseer, CoauthorCS, and Pubmed [39] are used. The train/val/test division ratio is 4:2:4. The details of these three data sets can refer to [39]. Given that all the data sets are multi-class, the noise synthesized manner used in Section 6.3 is followed.

The settings for some parameters, including learning and dropout rates follow the setting in [39]. Other parameter settings are detailed in the supplementary materials.

The competing results show that all the methods obtain similar accuracies and our method does not yield better results. The detailed results are shown in the supplementary materials due to lack of space. The partial reason may be because the graph structure among samples can alleviate the negative contribution of noisy labels. We leave further theoretical analysis as our future work.

6.5. Discussion

The competing results on the above three classification tasks indicate that the logit-based compensation method Logit-$l1$ achieves the overall best performance on the text sentiment analysis and image classification when simulated noises are added. Our proposed other methods, Label-$l1$ and Mixed hard/soft, also outperform the classical Bootstrapping loss-based method in most cases. On sentiment analysis, Logit-$l1$ outperforms the Base algorithm even when no simulated noises are added.

The absolute improvements on all experimental runs seem not significant even when the noisy rate is 20% and 30%. The partial reason is that the noises are randomly synthesized and the leveraged deep models may be capable of tolerating these kind of noises. In addition, all the involved methods are end-to-end without detecting noisy (and hard) samples in a pre-processing step. A two-step with noise detection in the first step may increase the accuracies more significantly.

7. Conclusions

This study reveals a widely used yet unexplored machine learning strategy, namely, compensating. Machine learning methods leveraging or partially leveraging compensating comprise a new learning paradigm called compensating learning. To solidify the theoretical basis of compensation learning, a systematic taxonomy is constructed on the basis of which to compensate, how to infer, and the compensation granularity. To demonstrate the universality of compensation learning, several existing learning methods are explained within our constructed taxonomy. Furthermore, three concrete compensation learning methods (*i.e.*, Logit-$l1$, Label-$l1$, and Mixed label) are proposed. Extensive experiments suggest that our proposed methods are effective in robust learning tasks.

As this is a pilot study, the established taxonomy is far from complete and many open problems still exist. In addition, a killing application of compensation learning remains lacking. We plan to further address these problems in the future study.

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, pages 41–48, 2009.

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019.

[3] Kuang-Yu Chang, Chu-Song Chen, and Yi-Ping Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In *CVPR*, pages 585–592, 2011.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] Jia Deng, Jonathan Krause, and Li Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, pages 580–587, 2013.

[6] Pedro A Forero, Vassilis Kekatos, and Georgios B Giannakis. Robust clustering using outlier-sparsity regularization. *IEEE Transactions on Signal Processing*, 60(8):4163–4177, 2012.

[7] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[8] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471, 2000.

[9] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

[10] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[11] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020.

[12] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8536–8546, 2018.

[13] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *ICCV*, pages 5138–5147, 2019.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[16] James Hong and Michael Fang. Sentiment analysis with deeply learned distributed representations of variable length texts. *Stanford University Report*, pages 1–9, 2015.

[17] Hidekata Hontani, Takamiti Matsuno, and Yoshihide Sawada. Robust nonrigid icp using outlier-sparsity regularization. In *CVPR*, pages 174–181, 2012.

[18] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, pages 5375–5384, 2016.

[19] Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *ICCV*, pages 3326–3334, 2019.

[20] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2304–2313, 2018.

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009.

[23] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NeurIPS*, volume 1, page 2, 2010.

[24] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *CVPR*, pages 5051–5059, 2019.

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[26] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[27] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, pages 3355–3364, 2018.

[28] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[29] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NeurIPS*, volume 26, pages 1196–1204, 2013.

[30] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[31] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4334–4343, 2018.

[32] Martin Slawski, Emanuel Ben-David, et al. Linear regression with sparsely permuted data. *Electronic Journal of Statistics*, 13(1):1–36, 2019.

[33] Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention. In *ICML*, pages 6234–6243, 2019.

[34] Chenglong Wang, Feijun Jiang, and Hongxia Yang. A hybrid framework for text modeling with convolutional rnn. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2061–2069, 2017.

[35] Yixin Wang, Alp Kucukelbir, and David M Blei. Robust probabilistic modeling with bayesian data reweighting. In *ICML*, pages 3646–3655, 2017.

[36] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Cheng Wu, and Gao Huang. Implicit semantic data augmentation for deep networks. In *NeurIPS*, 2019.

[37] Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, and Xiaoyan Zhu. Sentiment analysis by capsules. In *Proceedings of the 2018 world wide web conference*, pages 1165–1174, 2018.

[38] Zhen Wang, Guosheng Hu, and Qinghua Hu. Training noise-robust deep neural networks via meta-learning. In *CVPR*, pages 4524–4533, 2020.

[39] Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. In *ICLR*, 2019.

[40] Liang Zhao, Tianyang Zhao, Tingting Sun, Zhuo Liu, and Zhikui Chen. Multi-view robust feature learning for data clustering. *IEEE Signal Processing Letters*, 27:1750–1754, 2020.