Proceedings of the 2023 International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/27/20230177

An iterative loss correction method for deep learning with noisy labels

Jiajun Chen[1], Xiaolun Xu[2,4], Cheng Zheng[3]

[1]Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, 14627, United States of America
[2]Department of Marine Technology, Norwegian University of Science and Technology, Trondheim, 7050, Norway
[3]Ira A. Fulton Schools of Electrical Engineering, Arizona State University, Tempe Arizona, 85287, United States of America

[4]xuxiaolun2@gmail.com

Abstract. Noisy labelling is a prevalent issue in real-world data, often causing deep neural networks (DNNs) to overfit. Prior research in this area primarily relies on the accurate estimation of noise transition matrices, which is contingent on identifying anchor points in the clean data domain. However, current methods typically estimate the anchor points using information from the noisy labels, potentially resulting in poor estimation. In contrast, our novel method aims to enhance precision by developing an estimator that jointly learns the transition matrices and anchor points through iterative learning. Our approach is validated on the IMDB and MNIST datasets, proving to be more precise and effective than previous methods.

Keywords: weak-supervised learning, deep learning, loss correction, noisy labelling.

1. Introduction

The ability of deep neural networks (DNNs) to leverage large-scale datasets with accurate labelling has led to extraordinary success in a number of machine learning applications. As a result of employing non-expert sources to reduce the high costs involved with expert labelling, numerous real-world datasets are frequently produced using noisy labels. When DNNs are trained on these datasets, the introduction of tainted labels can drastically diminish the quality of the learnt models. Particularly, the existence of label noise can result in the overfitting of false patterns and poor generalization performance, especially when predicting complicated functions. Several methods the problem of noisy labelling have been studied in previous deep learning research. Estimating the noise transition matrix, which is employed in approaches such as loss correction [1-3], and noise adaption layer, is a prominent method. The matrices of transitions in noise provides information on the probability of ground truth label flipping under the assumption of instance-independent noise [4-6]. Transition matrices must be learnt accurately for these strategies to be successful. Finding anchor points in the clean data domain is a frequent approach for obtaining reliable estimates of the noise transition matrix. However, this method relies on calculating the noisy class posterior probabilities, which might lead to estimate mistakes. Considering this challenge, we intend to develop an improved iterative strategy that can increase the precision of identifying anchor

DOI: 10.54254/2755-2721/27/20230177

points and, therefore, the efficacy of inferring the matrix of transitions in Noise. This is how we've organized the rest of the material. In Section 2, similar work on deep learning with noisy labels is introduced. The third section covers the mathematical concept and our methods in depth. Section 4 presents our training outcomes, while Section 5 provides the conclusion.

2. Related work

Existing works on deep learning with noisy labels are briefly discussed here.

Robust loss function: These methods seek to enhance the loss function on the assumption that it is possible to get close to the optimal Bayes risk on clean data by updating the loss function successfully for noisy data [7]. Based on the results presented in [8], it can be concluded that the mean absolute error (MAE) loss is superior to the most popular categorical cross entropy (CCE) loss in terms of generalization. For large-scale data, however, the generalization performance of MAE loss diminishes. Subsequently, [9] presented the generalized cross entropy (GCE) loss, which is more resilient to noise and has the benefits of rapid convergence and great generalization capacity. Several expansions of GCE loss are based on Tsallis and Bregman divergences [10, 11]. Nevertheless, these strategies only work effectively in basic situations when the number of classes is modest.

Loss adjustment. Loss adjustment approaches try to limit the influence of noise on training data by modifying the loss of all samples according to their confidence levels. Based on their distinct adjustment mechanisms, these approaches may be categorized into four groups:

a) Loss correction: When attempting to repair forward or inverse loss, it is necessary to estimate a noise transition matrix [1]. Anchor points, which may be found using theoretical derivations or heuristics are often used to train this matrix properly [2]. However, T-revision has been created, a novel method for estimating the transition matrix without the need of anchor points. Using examples with a high noisy class posterior probability, the suggested method estimates the matrix, which is subsequently refined with the help of a slack variable. In a similar vein, presented instance-confidence embedding for computing the transition matrix to characterize the noise that is instance-dependent. In most cases, loss correction allows for a more in-depth analysis of the training data, and the success of loss correction methods is proportional to the accuracy with which the transition matrix is estimated [12,13].

b) Loss reweighting: To design a weighted training scheme, assigning a learnable weight to each sample in the noisy training dataset is necessary. With more weight given to correct labels and less to incorrect ones, these weights reflect how instructive the example is. By formulating a bilevel optimization problem, we may learn the example weights that serve as hyper-parameters for the main model. There are several common methods for calculating these weights, such as active bias, which gives inconsistent variances to label predictions, and significance reweighting, which uses the ratio between the distributions of noisy and clean data. In addition, Dual Graph takes into account the underlying structural connections between labels in Graph neural networks [14].

c) Label Refurbishment: Combining noisy and DNN-predicted labels with given confidence yields refurbished labels. Dynamic bootstrapping is a development of the original bootstrapping, which obtains the confidence of noisy labels through cross-validation, by adjusting a beta mixture model with two elements and one dimension to the loss distribution in real time. SELFIE is a new approach that selectively refines and corrects unclean or refurbished samples based on consistent label prediction and then combines these refurbished instances with the losses of clean samples for loss correction [15].

d) Meta Learning: The notion of learning to learn forms, the fundamental basis of meta learning, which entails the automatic adjustment of loss weighted functions or corrected labels. The CWS method uses a meta-DNN to help train the target DNN, and vice versa, while learning the weighted function [16]. At completion of the meta-DNN training with a pristine validation set, the weight score for the target DNN is evaluated. Automatic reweighting is another method that uses the updated gradient directions of training examples to reweight the tones of backward loss of mini-batch data, hence minimizing the loss of a pristine collection of validators [17]. Knowledge distillation is an approach to meta-learning for label refurbishment that uses the predictions of a specialized DNN that has been meticulously trained on a tidy validation dataset to train the target model. Unfortunately, these

DOI: 10.54254/2755-2721/27/20230177

techniques have the drawback of needing pristine validation data, which is sometimes difficult to get in practice [18].

3. Theorem support and mathematic notations

Assume s is a positive integer, [s] stands for the sets {1,2, 3, …, s}. If no addition explain is given, v is a column vector. M is a matrix, component in M with grid coordinates: i, j is denoted M_{ij}. Parameter i and j are integers from 1 to s, denote as i,j∈[s]. All-ones matrix is denoted $ones$.

In our noisy label data set with s classes, assume its feature space is χ, $x \in \chi$. And y represents the label space, $y = \{e^i : i \in [s]\}$. e^i stands for the ith linearly independent basis, which satisfies $e^i \in \{0,1\}^s, ones^T e^i = 1$. In our observations, samples with labels pairs (x, y) obeys a distribution probability $p(x, y)$. Our data set follows the distribution $p(x, y) = p(y|x)p(x)$ over space $\chi \times y$.

An n-layer neural network from $\chi \to \mathrm{R}^s$, is denoted by:

$$h = (h^{(n)} \bullet h^{(n-1)} \bullet h^{(n-2)} \bullet \ldots \bullet h^{(1)}).$$

For any $i \in [n-1]$, we have:

$$h^{(i)}(z) = \sigma(W^{(i)}z + b^{(i)})$$

When $i = n$:

$$h^{(n)}(z) = W^{(n)}z + b^{(n)})$$

$W^{(n)}$ and $b^{(n)}$ are unknown matrix and vectors we need to estimate. In this paper, σ uses RELU activatio n function. $\sigma(x)_i = \max(0, x_i)$. For convenience, we use equation:

$$x^{(i)} \approx h^{(i)}x^{(i-1)}$$

$x^{(0)} \approx x$, $h(x)$ signifies the weights assigned to each class by the model $argmax_{i\in[s]}h_i(x)$. Then, use two steps to compare it with true label Y. $p(y|x)$ to represent $x \in \chi$ $\tilde{p}(y|x)$ represents estimated probability distribution. $h_i(x)$ to get an approximating $p(y|x)$ $\tilde{p}(y|x)$, and then use loss function to measure the discrepancy. In this paper, every loss function forms a possible formula of:

$$\ell\big(\tilde{p}(y|x)\big) = (\ell(e^1, \tilde{p}(y|x)), \ldots, \ell(e^s, \tilde{p}(y|x)))^T \quad (1)$$

A significant proportion of the labels assigned to the training set in our data sets are vulnerable to noise. Thus, the true label remains undiscovered and unknown. In light of this constraint, we use sample distribution to conduct a comprehensive analysis of the dataset.

Theorem 1 suppose $\tilde{y}$ represents the observed label of sample, $p(x, \tilde{y})$ is the probability of noisy label $\tilde{y}$ distribution for each $x \in \chi$, $p(x, \tilde{y}) = \sum_y p(\tilde{y}|y)p(y|x)p(x)$, there exists a transition matrix M, for every integer i, j ∈ $[n]$, such that:

$$M_{ij} = p\big(\tilde{y} = e^i, y = e^i\big) \; \forall i, j \in n$$

If this matrix is known, the subsequent computations become straightforward. Thus, it becomes imperative to develop a reliable methodology for estimating M with a desirable degree of accuracy. A loss function in terms of cross entropy has been proposed, as shown in linked as equation (2):

$$\ell\left(e^i, \tilde{p}(y|x)\right) = -\left(e^i\right)^T log\tilde{p}(y|x) = -log\tilde{p}\big(y = e^i\big|x\big) \quad (2)$$

3.1. The first-step loss function procedure

Now, it is necessary to create an estimator of loss function ℓ by using the expected value of label noise, a short proof of its unbiased attributes are given as follows.

Theorem 2.Suppose that there is a non-singular matrix M, $\tilde{p}(y|x)$ represents estimated conditional probability distribution of true label, we define a new loss function estimator as:

DOI: 10.54254/2755-2721/27/20230177

$$\ell^1\big(\tilde{p}(y|x)\big) = M^{-1}\ell\big(\tilde{p}(y|x)\big)$$

On the right is an inverse cross-entropy loss function, with the expectation of ℓ when $x = x, y = \tilde{y}$ as $E_{x\&\tilde{y}}\|\ell$, then we have:

$$E_{x\&\tilde{y}}\|\ell^1\big(y, \tilde{p}(y|x)\big) = E_{x\&y}\|\ \ell(y, \tilde{p}(y|x))$$

Each side uses argmin function, it's clear that:

$$argmin(E_{x\&\tilde{y}}\|\ell^1\big(y, \tilde{p}(y|x)\big)) = argmin(E_{x\&y}\|\ \ell(y, \tilde{p}(y|x)))$$

PROOF:

$$E_{x\&\tilde{y}}\big\|\ell^1\big(\tilde{p}(y|x)\big) = E_{x\&y}\big\|\ \Big(M * \ell^1\big(\tilde{p}(y|x)\big)\Big) = E_{x\&y}\|\ \Big(M * M^{-1}\ell\big(\tilde{p}(y|x)\big)\Big) = E_{x\&y}\|\ \ell(\tilde{p}(y|x))$$

For the purpose of the new loss function, the loss values for each label are linearly combined. Analytical Matrix for Change M gives probability of each possible true label, akin to the functioning of a Markov chain. Typically, Theorem 1 indicates the invertibility of the Transition Matrix in most scenarios. However, in cases where M turns out to be singular, an alternate strategy is required to proceed with the task at hand. A possible solution involves utilizing a modified transition matrix $M' = \mu M + (1-\mu)I, \mu \in (0,1)$ which could be used in a more conservative noise-free set.

3.2. The second-step loss function procedure

In the following step, we need to start at a no loss function predicate results using neural network to get $\tilde{p}(\tilde{y}|x)$. Then, invoking Theorem 1 and Equation (2), alongside the cross-entropy loss function, yields the following expression:

$$\ell\left(e^i, \tilde{p}(y|x)\right) = -\big(e^i\big)^T log\tilde{p}(y|x) = -log\tilde{p}\big(y = e^i|x\big) \tag{3}$$

$$= -log\sum_{j=1}^{s} p\big(\tilde{y} = e^i, y = e^i\big)\tilde{p}\big(y = e^j|x\big) \tag{4}$$

$$= -log\sum_{j=1}^{s} M_{ji}\,\tilde{p}\big(y = e^j|x\big) \tag{5}$$

β analyze its behavior, we need to use definitions of broad loss family named proper composite. Next, define a link function β :

$$\ell_\beta(y, h(x)) = \ell\left(\mathrm{y}, \beta^{-1}\big(h(\mathrm{x})\big)\right) \tag{6}$$

An inverse link function predicted values which are on kinds of labels. When proper composite losses are applied, using argmin function to get less error result of the conditional probability distribution of true label $p(y|x)$:

$$\text{argmin}\ \mathrm{E}_{x\&y}\ell_\beta(y, h(x)) = \beta(p(y|x)) \tag{7}$$

Theorem 3 suppose that there is a non-singular matrix M, under the condition when proper composite losses ℓ_β, we define a new second loss function estimator as:

$$\ell_\beta^2(h(x)) = \ell(M^T\beta^{-1}(h(x)))$$

Similar as theorem 2, right hand side is a decline in cross-entropy loss formula, use the same denotes then we have:

$$E_{x\&\tilde{y}}\|\ell_\beta^2\big(y, h(x)\big) = E_{x\&y}\|\ \ell_\beta(y, h(x))$$

Each side uses argmin function, it's clear that:

$$argmin(E_{x\&\tilde{y}}\|\ell_\beta^2\big(y, h(x)\big)) = argmin(E_{x\&y}\|\ \ell_\beta(y, h(x)))$$

DOI: 10.54254/2755-2721/27/20230177

PROOF: if we denote $\alpha^{-1} = \beta^{-1}M^T$, then $\alpha = (M^{-1})^T\beta$, so original statement can be written as:

$$\ell_\beta^2 h(x) = \ell(M^T\beta^{-1}(h(x))) = \ell_\alpha(y(h(x))) \tag{8}$$

After this, ℓ_α can be deal with as a new proper composite loss. Using definition of proper composite loss (equation (7)), to minimize the loss over a noisy distribution, we have:

$$argmin(E_{x\&\tilde{y}}\|\ell_\alpha(y, h(x))) = \alpha(p(y|x)) \tag{9}$$

$$= \beta\big((M^{-1})^T p(\tilde{y}|x)\big) = \beta(p(\mathrm{y}|\mathrm{x})) \tag{10}$$

3.3. Optimized algorithm

Although the two preceding approaches perform effectively in a variety of scenarios, they both require the transition matrix M. In noisy label learning, however, this matrix is left unaltered and must be approximated. To solve this difficulty, we offer in this study a critical approach dubbed "anchor point" that serves as the foundation for our following optimization-based techniques.

Theorem 4 For each class $i \in [s]$, if the anchor point x^iexists, the two conditions below are met:

(1) $p(\bar{x}^i) > 0 \ \wedge p(y = e^i|\bar{x}^i) = 1$

(2) there are sufficient samples to model $p(\tilde{y}|x)$ accurately.

Combined these conditions with theorem 1, we can get:

$$M_{ij} = p(\tilde{y} = e^i|\bar{x}^i) \ \forall i, j \in [s]$$

PROOF: From assumptions above and notes we clarified, $p(\tilde{y}|x) = p(\tilde{y} = e^i|x)$, we can use $p(\tilde{y} = e^i|x)$ instead of $\tilde{p}(\tilde{y} = e^i|x)$, and for arbitrary $i, j \in [s]$, x is in feature vector space, we can get:

$$p(\tilde{y} = e^j|x) = \sum_{k=1}^{s} M_{kj}p(y = e^k|x) \tag{11}$$

We plug when $x = \bar{x}^i$, then get $p(y = e^k|\bar{x}^i) = 0$ if $k \neq i$.

The anchor point approach relies mostly on noisy label class probability estimations to determine each component of transition matrix M. Use, in other words, noise-contaminated labeled network output in SoftMax form. In particular, training set samples do not need to be labeled for this technique to make sense, if they follow the same distribution. And in our algorithm particularly, we use the two equations below:

Anchor Points:

$$\bar{x}^i = argmax_{x \in X}\tilde{p}(\tilde{y} = e^i|x) \tag{12}$$

Transition Matrix:

$$\widehat{M_{ij}} = \tilde{p}(\tilde{y} = e^i|\bar{x}^i) \tag{13}$$

With all this preparation, we can finally start using Theorem 4 to systematically train our sets. When we dig deeper, however, we find that there are several caveats associated with using anchor points for calculating the transition matrix. As it turns out, the connection between anchor points and transition matrices is not as straightforward as would first appear. In noisy label learning, the anchor points and transition matrix are affected by the lack of knowledge about the true label.

To sum up, the anchor points and the transition matrix are intertwined in a way that makes it risky to use one to estimate the other. We suggest adding an iterator to our method in order to increase the precision of our findings, and we draw inspiration from joint learning models and similar research.

Further investigation suggests that there may be a more suitable alternative. To begin, we develop an estimator to process noisy labels, coming from some clusters, in order to obtain labeled samples in good condition. After that, we may make educated guesses based on sample data.

DOI: 10.54254/2755-2721/27/20230177

$$\bar{x}^i = argmax_{x \in X} \tilde{p}(y = e^i|x) \quad (14)$$

instead of $\tilde{y}$. If it's possible, it can greatly improve our accuracy of transition matrix. And here is our algorithm:

Algorithm - optimized training

Input: Using noisy label data sets MNIST and MDB, separated into a test group and a practice group, ℓ uses default cross-entropy loss function, s is the class size, under this situation transition matrix is unknown.

use an estimator to get some clean data samples of some small clusters of classes $\{k_1, k_2, ...\}$, $k_1, k_2 \in [s]$.

train relative network weights matrix $h_1(x)$ on set MNIST and MDB of other classes with ℓ.

Combine two parts together and obtain a sample set X'.

Estimate transition matrix $\widetilde{M^{(0)}}$ by anchor points $(\bar{x}^i)^{(0)}$, for classes $\{k_1, k_2, ...\}$, using equation (14), for other classes, using equation (12) and (13).

Use transition matrix to estimate new anchor points $(\bar{x}^i)^{(1)}$,considering X' and $(\bar{x}^i)^{(1)}$to get transition matrix $\widetilde{M^{(1)}}$.

Finally, train network h(x) again using $\widetilde{M^{(1)}}$ on training sets of MNIST and MDB with ℓ^1 and ℓ_β^2.

Output: h (…)

As algorithm summarized, if transition matrix is known with high accuracy, all the work next can go smoothly. Otherwise, training the network requires a loss function ℓ on noisy classify information and assemble it using estimations of distribution $p(\tilde{y} = e^i|x)$ for most class. For some classes, we can find clean label data, using $p(y = e^i|x)$ directly to get anchor points.

4. Experiment

4.1. Datasets

4.1.1. MNIST. We conducted our analysis with data from the MNIST dataset, which contains 50k training examples and 10k test examples. The major goal of the collection is to identify the 10 different classes of objects inside the grayscale digital photographs of handwriting that measure 28 by 28 pixels [19]. (Ranging from 0 to 9). Experiments were done with a validation set comprised of 10 percent of the training data. Each and every piece of information in this dataset is accurate. Using the parameter matrix T, error structures such as "DEER" and "HORSE" are emulated by intentionally corrupting the labels. Sym-20 label noise means 20% of cases have noisy labels, whereas Sym-50 label noise shows 50% of labels are noisy. To make the model more resistant to class-conditional noise, we choose for Class-conditional Noise (CCN) . We find that the properties of the raw dataset are better reflected by the CNN setup. In the real-world photo labelling challenge, the divergence of human subjective cognition on the

DOI: 10.54254/2755-2721/27/20230177

likelihood of mislabelling as "car" means that a picture with the true label "cow" will have a considerably higher chance of being mislabelled as "donkey."

4.1.2. IMDB. For testing how well our strategy works in the real world, we turn to the IMDB dataset [20]. Up to 30 reviews can be included for each movie in the dataset, which totals 100,000 reviews total. Reviews of movies often use question marks, exclamation points, and exclamation marks. In all, there were 88,585 unique terms found in the reviews used to populate the dataset [20]. Fifty thousand were labelled and used for testing; for the remaining fifty thousand, a negative review should be represented by a zero and a good review by a one. With this method, a critic may provide a rating out of 10 to a selection of films. Each batch of labelled reviews consists of 12,500 evaluations, including 12,500 good evaluations and 12,500 negative evaluations. The procedure of iteratively transforming Y with noisy labels into an estimate of the precise labels Y through the transition matrix T is repeated until an accurate estimate is reached. We think that by repeating these simple steps over and over again, we may eventually arrive at the fundamental truth Y.

4.2. Implementation

We developed a deep learning system to train on the IMDB and MNIST datasets and included it in this post. Our studies were similar in using a matrix T to iteratively approach the true value Y by converting class vectors to binary class matrices. We used a generalized table entropy loss during training on the network and compared the actual and expected values to investigate the impact of learning rate (lr) and noise rate (nr) on the loss of learning ability and test accuracy. For evaluating verification accuracy in a noisy environment, this approach becomes useful. We used ReLU for all networks, initialized the weights before ReLU, and set the number of batch rounds to 128. Apply an estimator of the matrix T of noise labels to X' as the training and validation set. Preliminary experiments demonstrated that band iterations significantly improved the Y approximation after multiple iterations.

4.3. Experiments on MNIST

To evaluate the method's performance on noisy tags, we train a network using the method on MNIST. (Noise rate r = 0.0, 0.1, 0.2, 0.3). In the experiments, we consider MNIST. We use a dropout probability of 0.5 when training an architecture with two 128-layer dense hidden layers. By "accuracy," we mean the total accuracy of the training set, and by "val-accuracy," we mean the total accuracy of the validation set. To do this, we employ a baseline proficiency level 0.01 and run for 40 epochs and $\delta = 10^{-6}$ considering the effects of noise and weight initialization. The number of iterations for each experiment is determined to stop when the accuracy of the last two training sets is less than 0.01. As shown in Figure 1, slight noise does not affect the model's accuracy. The precision achieved via forward correction is depicted in Figure 1, whereas the precision achieved through reverse correction is depicted in Figure 2. On the whole, our technique outperforms MINST as regards the reliability of the tests and recovery accuracy. Accuracy on tests improves up to a maximum of 40 iterations when training with big labels at the outset. Adapting the network to noisy labels is made easier by decreasing the learning rate. At least until the very end of training, our iterative optimization strategy maintains a high-test accuracy. This is a crucial by product of the iterative optimization process we employ.

DOI: 10.54254/2755-2721/27/20230177

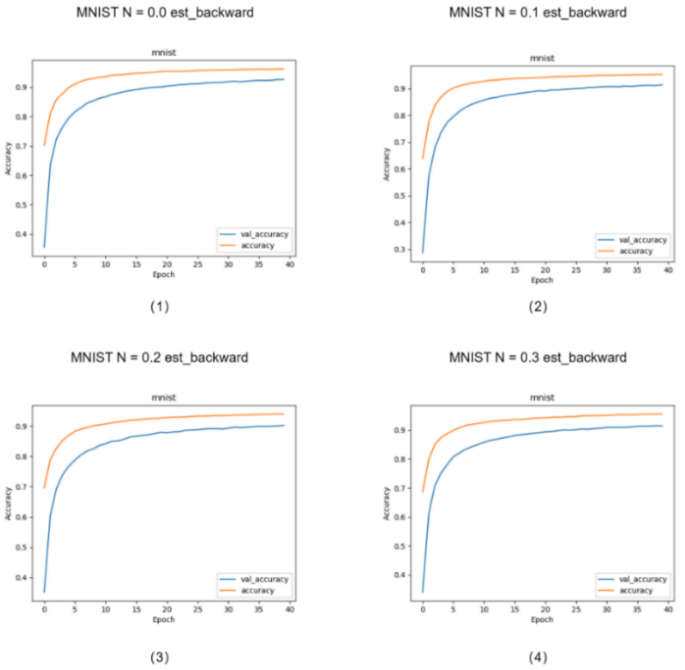

Figure 1. MINST Est backward.

In Figure 1, We employ a loss function based on the cross entropy to conduct experiments on the MINST data set with a noise ratio of (0.0, 0.1, 0.2, 0.3), and the figure 1 shows the precision of the entire training dataset and the overall veracity of the validation sample accuracy (val_accuracy). In Figure 2, We employ a loss function based on the cross entropy to conduct experiments on the MINST data set with a noise ratio of (0.0, 0.1, 0.2, 0.3), and the Figure 2 displays the general precision. of the forward-corrected practice drills (accuracy) and the validation sample set the precision as a whole (val_accuracy).

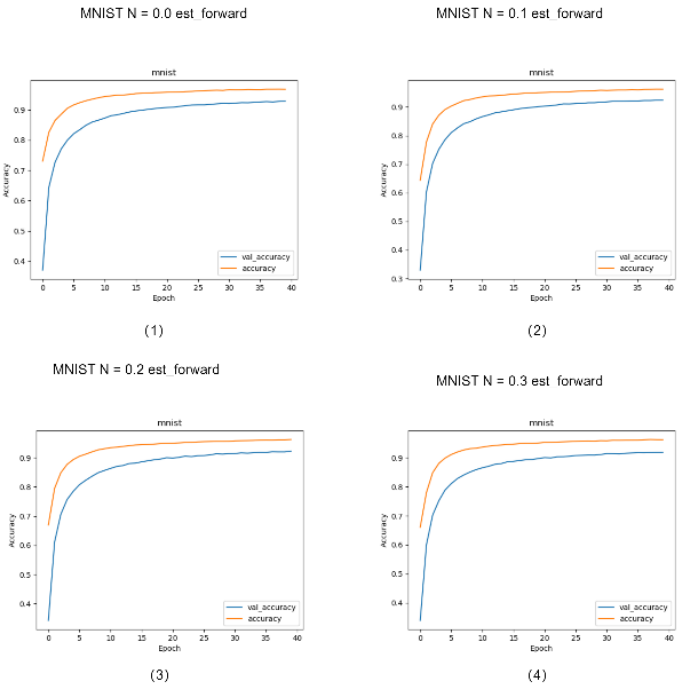

Figure 2. MINST Est forward.

DOI: 10.54254/2755-2721/27/20230177

In Table 1, we record the experimental data on the MINST dataset with the noise ratio (0.0, 0.1, 0.2, 0.3) respectively. Forward-corrected training set accuracy (accuracy), validation set accuracy (val accuracy), deficit in training sets (loss), validation set loss (val loss), and Precision in testing sets (accuracy) are all displayed in the figure (Test Accuracy).

Table 1. MNIST Est forward.

MNIST, Est forward, Fully Connected				
Method	Test Accuracy (%)			
Noise Rate (%)	0	10	20	30
Accuracy	0.9282	0.9239	0.9224	0.9187
Val_accuracy	0.9675	0.9620	0.9622	0.9622
Loss	0.4826	0.5416	0.5703	0.6536
Val_loss	0.4187	0.4773	0.5035	0.5920
Test Accuracy	0.9261	0.8951	0.9124	0.8975

In Table 2, we record the experimental data on the MINST dataset with noise ratio (0.0, 0.1, 0.2, 0.3) respectively. The figure depicts the training of backward correction, the precision of the entire training data set (accuracy), the precision of the validation sample as a whole (val accuracy), the actual loss value of set used for training (loss), and the exactness of the test battery as a whole (Test Accuracy).

Table 2. MNIST Est backward.

MNIST, Est backward, Fully Connected				
Method	Test Accuracy (%)			
Noise Rate (%)	0	10	20	30
Accuracy	0.9298	0.9270	0.9156	0.9136
Val_accuracy	0.9502	0.9630	0.9502	0.9475
Loss	0.3731	0.4302	0.4178	0.4012
Val_loss	0.3012	0.3580	0.3342	0.3129
Test Accuracy	0.9011	0.8848	0.8695	0.8934

4.4. Experiments on IMDB

After training the network with the IMDB technique (r = 0.0, 0.1, 0.2, 0.3 for the noise ratio), we assess its efficacy. Starting with a learning rate of 0.01 and in sum of 5000 randomly picked features, we run the algorithm for 60 iterations. Passing requires training a model to assign 50-dimensional embeddings to words. Using ReLU, the output of the embedding is subjected to a dropout with a probability of 0.8. Interestingly, the iterative method vastly enhances the Y approximation from the ground truth. The forward correction for 0-30% noise is depicted in Figure 3, while the reverse correction is depicted in Figure 4. The above example shows that our model is insensitive to illumination fluctuations. On IMDB, we observed perfect noise robustness and perfect forward correction.

DOI: 10.54254/2755-2721/27/20230177

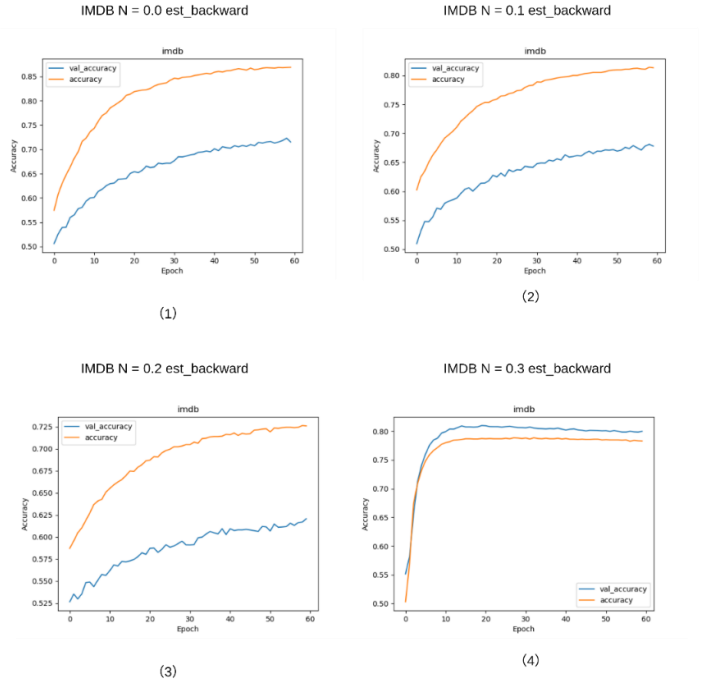

Figure 3. IMDB Est forward.

In Figure 3, The picture displays the overall reliability of the backward corrected practice sets (accuracy) and the precision of the validation sample as a whole (val accuracy) by use of the relu of cross-entropy on the IMDB data set with a noise ratio of (0.0, 0.1, 0.2, 0.3).

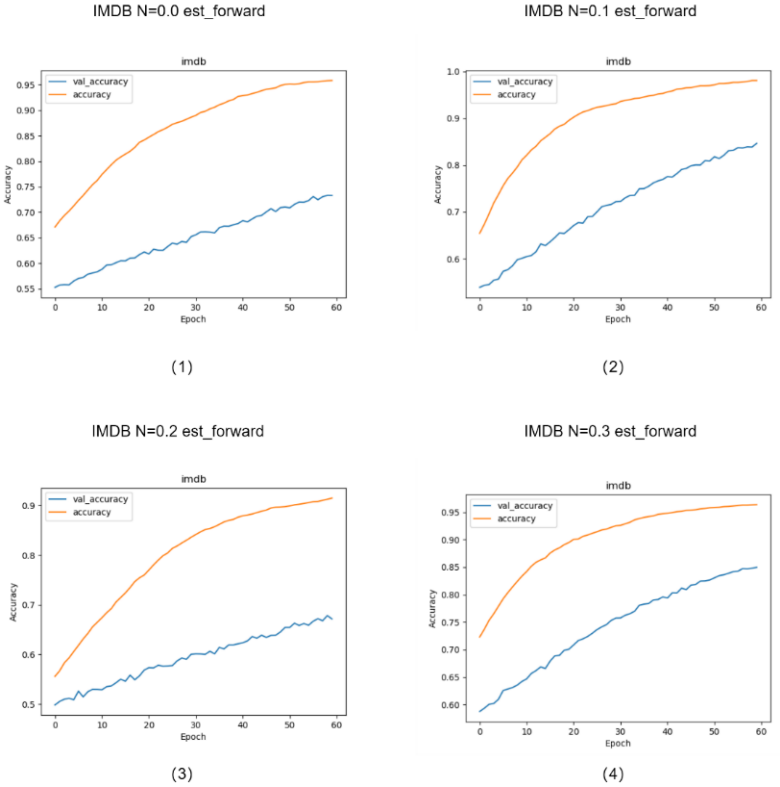

Figure 4. IMDB Est backward.

DOI: 10.54254/2755-2721/27/20230177

In Figure 4, The image depicts the overall reliability of the forward-corrected practice sets (accuracy) and the precision of the validation sample as a whole (val accuracy) by use of the relu of cross-entropy on the IMDB data set with a noise ratio of (0.0, 0.1, 0.2, 0.3). In Table 3, we record the experimental data on the MINST dataset with the noise ratio (0.0, 0.1, 0.2, 0.3) respectively. Backwards-correction training is depicted together with the set's overall accuracy (accuracy), the precision of the validation set as a whole (val accuracy), the training set's overall loss value (loss), validating samples' overall loss value (val loss), and the test set's overall accuracy (accuracy) (Test Accuracy).

Table 3. IMDB Est backward.

IMDB, est_forward , Fully Connected				
Method	Test Accuracy (%)			
Noise Rate (%)	0	10	20	30
Accuracy	0.6921	0.5949	0.6322	0.7230
Val_accuracy	0.9076	0.7596	0.8336	0.8944
Loss	0.6931	0.6932	0.6930	0.6930
Val_loss	0.6932	0.6932	0.6930	0.6930
Test Accuracy	0.7013	0.6933	0.6932	0.6532

In Table 4, we record the experimental data on the MINST dataset with noise ratios of (0.0, 0.1, 0.2, 0.3) respectively. Forward corrective training is depicted in the picture along with correctness of the set as a whole (accuracy), precision in the validation set (val accuracy), reduction in training sets (loss), drawback in the validation set (val loss), and validity of the test samples (accuracy) and the overall accuracy of the test set (Test Accuracy).

Table 4. IMDB Est forward.

IMDB est_backward , Fully Connected				
Method	Test Accuracy (%)			
Noise Rate (%)	0	10	20	30
Accuracy	0.9614	0.9598	0.9686	0.9652
Val_accuracy	0.9614	0.9712	0.9884	0.9732
Loss	0.4463	0.3954	0.3651	0.3510
Val_loss	0.4212	0.3714	0.3572	0.3510
Test Accuracy	0.6327	0.6007	0.5756	0.5481

5. Conclusion

Trained on massive datasets, deep neural networks (DNNs) have demonstrated impressive accuracy for picture categorization. People may quickly gather large-scale datasets from the web. However, they frequently have incorrect labels, also called "noisy labels." We suggest a novel iterative approach to enhancing the reliability of the model. Cross entropy is also utilized in our loss function. This technique can prevent the activation function from entering the saturation area under certain conditions when the

DOI: 10.54254/2755-2721/27/20230177

error is significant, the gradient is large, and the decrease is rapid. These statistics are honed using the noisy IMDB and MINST datasets for training. Our research ensures the framework's efficacy; it may be taught in noisier environments without compromising model precision. The findings demonstrate that our approach is somewhat predictive.

References

[1] Patrini, Giorgio, et al. "Making deep neural networks robust to label noise: A loss correction approach." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[2] Liu, Tongliang, and Dacheng Tao. "Classification with noisy labels by importance reweighting." IEEE Transactions on pattern analysis and machine intelligence 38.3 (2015): 447-461.

[3] Scott, Clayton. "A rate of convergence for mixture proportion estimation, with application to learning from noisy labels." Artificial Intelligence and Statistics. PMLR, 2015.

[4] Chen, Xinlei, and Abhinav Gupta. "Webly supervised learning of convolutional networks." Proceedings of the IEEE international conference on computer vision. 2015.

[5] Bekker, Alan Joseph, and Jacob Goldberger. "Training deep neural-networks based on unreliable labels." 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016.

[6] Sukhbaatar, Sainbayar, et al. "Training convolutional networks with noisy labels." arXiv preprint arXiv:1406.2080 (2014).

[7] Manwani, Naresh, and P. S. Sastry. "Noise tolerance under risk minimization." IEEE transactions on cybernetics 43.3 (2013): 1146-1151.

[8] Ghosh, Aritra, Himanshu Kumar, and P. Shanti Sastry. "Robust loss functions under label noise for deep neural networks." Proceedings of the AAAI conference on artificial intelligence. Vol. 31. No. 1. 2017.

[9] E. Clarke, O. Grumberg, S. Jha, et al., Counterexample-guided abstraction refinement, in: E.A. Emerson, A.P. Sistla (Eds.), Computer Aided Verification, Springer, Berlin, Heidelberg, 2000, pp. 154–169. DOI: https://doi.org/10.1007/10722167_15

[10] Zhang, Zhilu, and Mert Sabuncu. "Generalized cross entropy loss for training deep neural networks with noisy labels." Advances in neural information processing systems 31 (2018).

[11] Amid, Ehsan, et al. "Robust bi-tempered logistic loss based on bregman divergences." Advances in Neural Information Processing Systems 32 (2019).

[12] Xia, Xiaobo, et al. "Are anchor points really indispensable in label-noise learning?." Advances in neural information processing systems 32 (2019).

[13] Zhang, Yivan, and Masashi Sugiyama. "Approximating instance-dependent noise via instance-confidence embedding." arXiv preprint arXiv:2103.13569 (2021).

[14] Wang, Ruxin, Tongliang Liu, and Dacheng Tao. "Multiclass learning with partially corrupted labels." IEEE transactions on neural networks and learning systems 29.6 (2017): 2568-2580.

[15] Reed, Scott, et al. "Training deep neural networks on noisy labels with bootstrapping." arXiv preprint arXiv:1412.6596 (2014).

[16] Dehghani, Mostafa, et al. "Avoiding your teacher's mistakes: Training neural networks with controlled weak supervision." arXiv preprint arXiv:1711.00313 (2017).

[17] Ren, Mengye, et al. "Learning to reweight examples for robust deep learning." International conference on machine learning. PMLR, 2018.

[18] Li, Yuncheng, et al. "Learning from noisy labels with distillation." Proceedings of the IEEE international conference on computer vision. 2017.

[19] Hendrycks, Dan, et al. "Using trusted data to train deep networks on labels corrupted by severe noise." Advances in neural information processing systems 31 (2018).

[20] A. Maas, R. Daly, P. Pham, D. Huang, A. Ng, and C. Potts, “Learning Word Vectors for Sentiment Analysis,” ACL, 2011.