Multimedia Tools and Applications (2022) 81:13593–13610
https://doi.org/10.1007/s11042-022-12323-2

1176: ARTIFICIAL INTELLIGENCE AND DEEP LEARNING FOR BIOMEDICAL APPLICATIONS

BAW: learning from class imbalance and noisy labels with batch adaptation weighted loss

Siyuan Pan[1] · Bin Sheng[1] · Gaoqi He[2] · Huating Li[3] · Guangtao Xue[1]

Received: 31 July 2020 / Revised: 15 October 2021 / Accepted: 17 January 2022 /
Published online: 21 April 2022

Abstract

Deep learning has made significant achievements in the field of medical image processing. To train a robust model with strong generalization, a large-scale, high-quality dataset with balanced categories and correct labels is required. However, most datasets follow a long-tail distribution that some classes occupy most of the data, and other classes have only a few samples. At the same time, incorrect labels exist in the datasets. The existing methods focus on solving only one of these two problems, such as Focal Loss for class imbalance and mean-absolute error loss function for noisy labels. However, methods that try to alleviate one of the problems will aggravate the other. In order to tackle the class imbalance while avoids fitting the noisy labels, we propose a novel Batch Adaptation Weighted (BAW) loss. It uses the loss weights of known samples to guide the direction of network optimization for next batch training. BAW is easy to implement and can be extended to various deep networks to improve accuracy without any extra cost. We evaluate BAW on a general natural image dataset, CIFAR-10, and verify it on a large-scale medical image dataset, ChestX-ray14. Compared with existing algorithms, BAW gets best results on both datasets. Experiments shows that our algorithm can solve the problem of class imbalance and noisy labels at the same time. The code of our project is available at https://github.com/pansiyuan123/chestnet.

Keywords Class imbalance · Noisy labels · Batch adaptation weighted · ChestX-ray14

✉ Bin Sheng
binsheng@sjtu.edu.cn

✉ Huating Li
huarting99@sjtu.edu.cn

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

[2] East China Normal University, Shanghai 200241, China

[3] Shanghai Jiao Tong University Affiliated Sixth Peoples Hospital, Shanghai 200233, China

1 Introduction

With the emergence of large-scale data and the rapid improvement of computing power, deep learning method gradually gains the ability to surpass human beings in many visual tasks [18, 27, 29], especially in the field of medical image processing [34].

Usually, a simple network can handle common visual tasks (such as classification, segmentation, detection) well. In this paper, we proposed a network based on DenseNet121 which can be shown in Fig. 1. However, such remarkable success is undoubtedly inseparable from the high-quality large-scale datasets [22, 28, 37]. However, most public datasets are not evenly distributed. Naturally, common categories have more samples and the rare types have less, because the difficulty of obtaining pictures is different [23].

Simultaneously, the compromise between label accuracy and dataset capacity has been an obstacle that limits the accuracy of deep learning models. Datasets such as social tagging and crowd-sourcing are noisy but easy to get. As a result, their scale can be massive. In comparison, the capacity of well collected and labeled datasets is limited. Moreover, even the well-labeled datasets can contain noise because of accidental label error, confusing and low-quality images, or reporting bias [24, 32]. Since the noise in the dataset will have a significant impact on the performance of the deep learning model, it is the key for the network to identify and modify the noise samples. The noisy label refers to the false ground truth given by the dataset (including training set and test set). This can be caused by many reasons. For example, the data set is obtained by crawler from internet without cleaning; The person who give the data labels has not been checked and confirmed after labeling; For some medical image data, the labeled person need some relevant experience. And we show some samples of these two datasets in Figs. 2 and 3.

The class imbalance will lead the network towards overfitting by the majority class and ignore the minority class [2], while noisy labels will cause the network to over-learn the wrong labels, which will both greatly reduce the performance of the model [30]. Recently, many works focus on solving one of the two problems, such as: [3, 23, 33] for class imbalance and [25, 31] for noisy labels. However, the tough issue is that the solutions to these two problems are incompatible. The loss function of the former (i.e., Focal loss [22]) pay more attention to the hard examples so their weights will be increased which can easily lead to overfitting to the noisy data. The solution to the latter (i.e., O2U-Net [13]) hopes to find a balance between loss under-fitting and over-fitting. while at this time, the network is likely only to learn to fit the categories with huge samples but under fit the categories with few samples. In conclusion, both of the above algorithms cannot solve the two problems at the same time so that the network has not perfect performance.

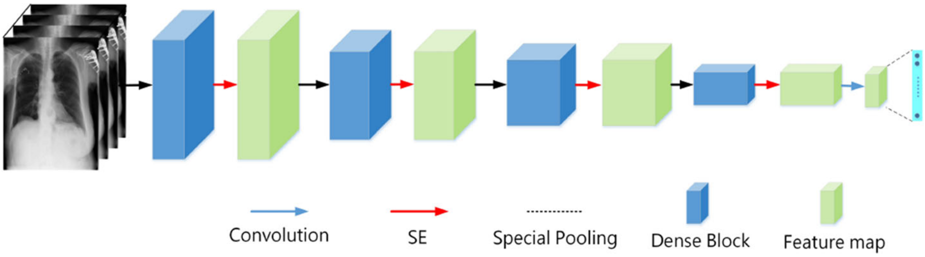


Fig. 1 The architecture of the proposed ChestNet is based on DenseNet121. ChestNet selects features from networks in channels and space through Squeeze-and-Excitation module [14] (SE) and Spatial pooling to solve the diversity and relevance of diseases. BAW loss is used to solve the class imbalance and to avoid the network overfitting with noisy labels

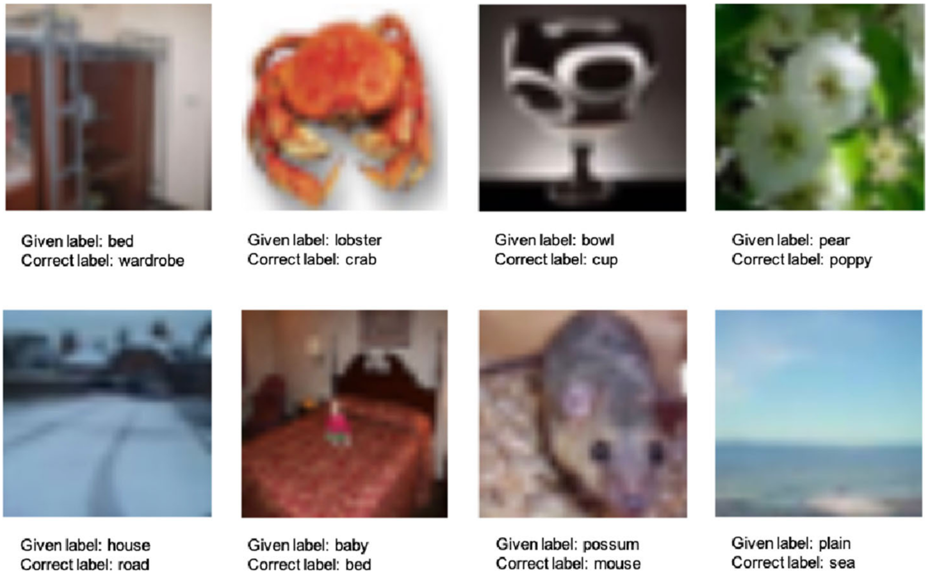


Fig. 2 Examples of noisy labels selected from CIFAR dataset [17]. The given label is the ground truth from the dataset. The correct label is the results computed by cleanlab [24]. Cleanlab is a data-centric python package for machine learning with noisy labels. cleanlab cleans labels and supports finding, quantifying, and learning with label errors in datasets

To solve the dilemmas mentioned above, this paper proposes a novel loss term called BAW, which uses the area under ROC (AUROC [7]) of the known samples to guide the weight of different classes for the next batch training. By doing so, BAW focus more on the smaller classes, thus solves the problem of class imbalance. Meanwhile, the low weight of well-trained categories will neglect the noisy label to avoid over-fitting. That is why BAW can deal with both problems at the same time. For detailed functional analysis of BAW, please refer to Section 5.3.2. It is easy-to-implement and can be extended to various deep neural networks.

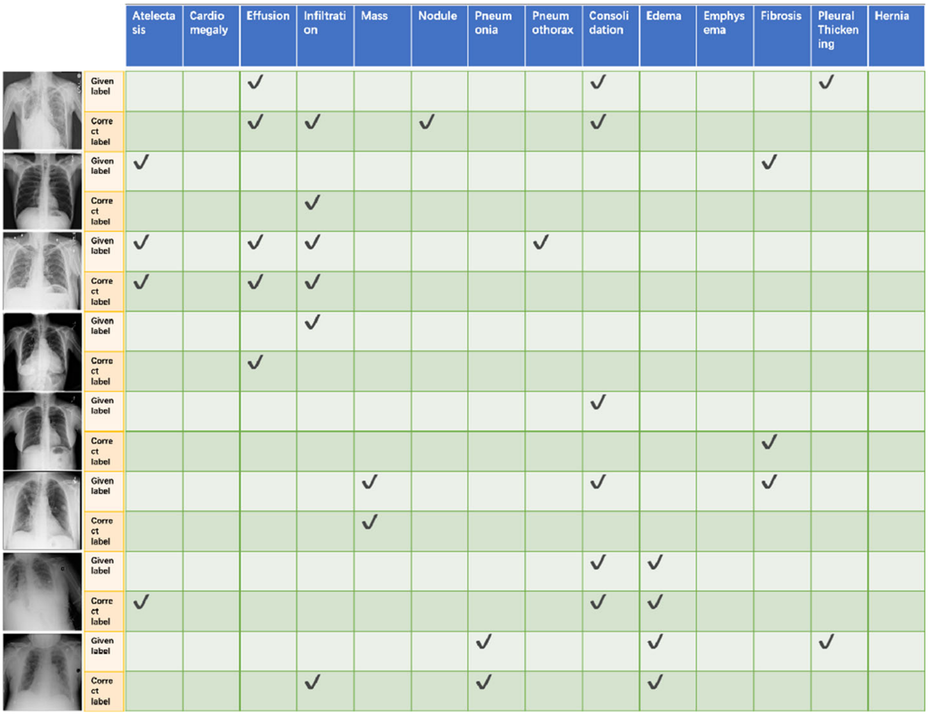
The contributions of this work can be summarized as follows:

- We propose a new loss function called batch adaptation weighted loss to solve the unbalanced distribution of categories. We uses the area under ROC (AUROC) of the known samples to guide the weight of different classes for the next batch training.
- In order to avoid excessive noisy samples in one batch, we use Exponential Moving Average (EMA) smoothing weights to obtain more stable class weights, which is proved useful during the experiment.
- We evaluate BAW on a general natural image dataset, CIFAR-10 [17], and verify it on a large-scale medical image dataset, ChestX-ray14 [34]. Our algorithm achieves best results on both datasets.

2 Related work

2.1 Class imbalance

Class imbalance is often encountered in the real dataset [3, 23, 33]. The solutions to this problem are mainly divided into two types, data preprocessing and loss function weighting.



	Atelectasis	Cardiomegaly	Effusion	Infiltration	Mass	Nodule	Pneumonia	Pneumothorax	Consolidation	Edema	Emphysema	Fibrosis	Pleural Thickening	Hernia
Given label			✓						✓				✓	
Correct label			✓	✓		✓			✓					
Given label	✓											✓		
Correct label				✓										
Given label	✓		✓	✓				✓						
Correct label	✓		✓	✓										
Given label				✓										
Correct label			✓											
Given label									✓					
Correct label												✓		
Given label					✓				✓			✓		
Correct label					✓									
Given label									✓	✓				
Correct label	✓								✓	✓				
Given label							✓			✓			✓	
Correct label				✓			✓			✓				

Fig. 3 Examples of noisy labels selected from chestX-ray14 dataset. The given label is the ground truth from the dataset. The correct label is the results computed by cleanlab [24]

Data preprocessing, such as undersampling [23], oversampling [3] and data enhancement, can solve the problem of class imbalance at the dataset level. However, due to poor operability, it is easy to cause data loss and difficult to use for all tasks. By comparison, loss function weighting can be easily applied to all tasks, and the modification is straightforward and convenient. At present, Focal loss [22] is proposed for the imbalance of positive and negative samples in detection, but this method is easy to overfit outliers, leading to the decline of model performance. Li et al. [20] proposed Gradient Harmonizing Mechanism (GHM) loss to conduct weight reduction processing. It makes the model focus more on general samples and discard difficult samples that containing a large amount of gradient information. However, Focal loss pays more attention to the difficult samples so the weights of them will be increased. This can easily lead to overfitting of noisy data.

2.2 Noisy labels

Learning with noise has been extensively studied in machine learning. Most of them includes noise-robust algorithms, noise-cleaning algorithms, noise modeling for learning with noisy label.

2.2.1 Noise robust methods

Many studies have shown that label noise can impact the classification accuracy of trained classifiers. To reduce the impact of noisy labels, some approaches rely on training classifiers using noise robust loss functions [8]. These methods alter normally used loss so that the new loss punish less on samples whose label is not correct. These methods is relatively simple, controllable and predictable. But most of them is not completely robust to label noise, as they reduce the impact of label noise but do not get rid of it. Some approaches enhance the network by regularization [4]. Applying regularization to the model can limit the search space for optimization and delay the over-fitting. Most methods of this kind are easy to implement and computing efficient. But the impact of regularization on deep network is hard to explain and predict. There are also methods that clean the dataset by remove samples or change labels based on training result. But these methods mix hard examples with mislabeled ones, which can be harmful especially when the number of labels is large and label distribution is not balanced. Most noise robust methods work under the assumption that the label noise can be avoided by not over-fitting the training data, which is not true for most datasets.

2.2.2 Semi-supervised learning

Semi-supervised learning is proposed to fully utilize weak and noisy labels, or even unlabeled dataset with well labeled data. Ding et al. [5] proposed a two-stage framework for learning with noisy labels. By ignoring labels that may be wrong, the algorithm avoids the noisy samples while utilizing the whole dataset. Some other methods employ label propagation and iterative label, and correct noisy dataset using the network trained on clean dataset. Most semi-supervised methods require complex calculation of image similarity or extra clean sub-dataset, which limits the usage of these methods. Moreover, semi-supervised learning relies heavily on the initial model trained on clean dataset, which is not robust under high noise ratio.

2.2.3 Transfer learning

Many previous studies have proved that CNNs are capable of learning rich and hierarchical visual features given that the training data is sufficient. Under the assumption that most visual features are common between different domains, researchers proposed to first initializing CNN parameters with a model pre-trained on a large scale dataset. The pre-trained network extract more useful features compared to random initialized network. If the two tasks are related, the pre-trained network will generate representative features and fine-tuning network on noisy dataset can be more robust compared to training the network from scratch.

2.2.4 Noise modeling

Many previous studies in this area focus on modeling the label noise [15, 24, 35]. These methods try to learn the joint distribution of given label and latent true label. In [15] a none-linear noise module (NNAQC) is appended to standard CNNs, and the loss in this work is designed to encourage clustering. The output of NNAQC is interpreted as a probabilistic model and performs as an efficient denoise model. Jiangchao [35] proposed a probabilistic model which explicitly introduce an extra variable to represent the trustworthiness of noisy labels. The confident learning (CL) method [24], however, works on the prediction of well trained networks. The CL module takes as input the predictions and corresponding labels, and estimate the joint distribution of label noise. These noise modeling models are efficient estimators for label noise distribution and can be used to clean the dataset, estimating prediction confidence, find the misleading labels in dataset. But almost all of these works focus on labels-wise distributions rather than instance-wise distributions. The images in these methods are divided into groups by given label and true label rather than the characters of images such as lighting, blurred or not, clarity, et.

2.3 Deep model on ChestX-ray14

Rajpurkar et al. [26] proposed CheXNet, which used the latest deep convolutional neural network structure Densenet [12] as the feature extraction module. The model uses heatmaps to obtain disease areas on the chest x-ray image. Li et al. [19] proposed a limited-supervised method combining disease classification and detection, using a small number of detection boxes to improve the identification and localization ability of chest diseases. However, class imbalance and noisy labels are not take into account in their method, so our method behaves better during the experiment.

3 Batch adaptation weighted loss

3.1 Inspiration

To solve the problem of class imbalance, the fundamental idea is to set the bigger categories' weight as a smaller number. That is what α does in focal loss [22]. However, this requires manual adjustment of the hyperparameters and will lead to over-learning of noisy labels. Therefore, we hope to make the network adjust the weights of different categories during the network training process adaptively. Based on this idea, we proposed a new loss function termed BAW. It uses the samples of multiple batches that have learned before and then to get the learning degree of each category, which is measured by AUROC. Use this as a guide to calculate the new weight W_i for each category i. At the same time, in order to solve the problem that the network is trapped in the noisy labels, we use Exponential Moving Average (EMA) [10] to obtain a more stable class weights.

3.2 Definition

We use $t \in T$ represents T iterations. $k \in K$ indicates that the batch size of each iteration is K. $f_{\Theta_t}(x)$ is the CNN mapping function when the parameter is updated to Θ_t after t iterations. At this time, the network learned a total of $T \times K$ samples after T iterations. We hope that the network can use the information of known samples to guide the next batch calculation. Let $\Lambda_i(x) \in [0, 1]$ denote the indicator function (*e.g.,* AUROC), where i represents the ith category. Here, the weight indicator of category i is expressed as $1 - \Lambda_i(\{f_{\Theta_t}(x)\}_{t,k})$. If the indicator of the category i is reduced after the T iterations, it indicates that the category i does not obtain sufficient update information in the T iterations. So the learning of the category i will be aggravated, and greater weight should be given. On the contrary, if the index of the category i is high enough, it indicates that the category i has been fully studied after T iterations compared with other categories. At this time, the focus of learning should be on other categories, so less weight should be given on category i. Finally, we use softmax function to smooth the weight of each category so that $\sum_i W_i = 1$. Here, the formula of weight indicator is

$$w_i = softmax(1 - \Lambda_i(\{f_{\Theta_t}(x)\}_{t,k})). \tag{1}$$

Furthermore, since the adaptive weight calculation in the formula comes from multiple mini-batches of sample indicators, in order to avoid the possibility of excessive noisy samples in the multi-batch sample, We uses EMA smoothing weights to obtain more stable class weights. We define w_i' as the weight factor of category i before updating. The update formula for w_i can be expressed as

$$w_i = \alpha w_i' + (1-\alpha) softmax((1 - \Lambda_i(\{f_{\Theta_t}(x)\}_{t,k}))). \tag{2}$$

The α here is used for balancing the new calculated weight and the old one. In experiments, a relatively small value (e.g., $\alpha = 0.1$, our default) works better than a big value (e.g., $\alpha = 0.5$).

So our loss function is:

$$L = -\frac{1}{N}\sum_{i=1}^{N} w_i (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})) \tag{3}$$

Here, we denote $y^{(i)}$ as the label of the instance, $\hat{y}^{(i)}$ is the prediction of our model. If we take the derivative of this loss function with respect to θ_j, we get

$$\begin{aligned}\frac{\partial L}{\partial \theta_j} &= -\frac{1}{N}\left(\frac{\partial w_i}{\partial \theta_j}\left(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})\right.\right.\\ &\qquad \left.\left. + \frac{\partial (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})}{\partial \theta_j} w_i\right)\right).\end{aligned} \tag{4}$$

First, we calculate the derivative of w_i for back propagation.

$$\begin{aligned}\frac{1}{N}\frac{\partial w_i}{\partial \theta_j} &= \frac{1}{N}\frac{\partial \alpha w_i' + (1-\alpha)\mathrm{softmax}((1 - \Lambda_i(\{f_{\theta_t}(x)\}_{t,k})))}{\partial \theta_j}\\ &= \frac{(1-\alpha)}{N}\frac{\mathrm{softmax}((1 - \Lambda_i(\{f_{\theta_t}(x)\}_{t,k})))}{\partial \theta_j}\\ &= \frac{(1-\alpha)}{N} \times \begin{cases} \Lambda_i(\{f_{\theta_t}(x)\}_{t,k})_i(1-\theta_j), i = j \\ -\theta_j \Lambda_i(\{f_{\theta_t}(x)\}_{t,k})_i, i \neq j \end{cases}\end{aligned} \tag{5}$$

Then we calculate the derivative of cross entropy for back propagation.

$$\begin{aligned}\frac{1}{N}\frac{\partial (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j}\left(\frac{1}{N}\sum_{i=1}^{N}[\log(1 + e^{\theta^T x^{(i)}}) - y^{(i)}\theta^T x^{(i)}]\right)\\ &= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial \theta_j}\log(1 + e^{\theta^T x^{(i)}}) - \frac{\partial}{\partial \theta_j}(y^{(i)}\theta^T x^{(i)})\right]\\ &= \frac{1}{m}\sum_{i=1}^{m}\left(\frac{x_j^{(i)} e^{\theta^T x^{(i)}}}{1 + e^{\theta^T x^{(i)}}} - y^{(i)} x_j^{(i)}\right)\\ &= \frac{1}{N}\sum_{i=1}^{N}(\hat{y}^{(i)} - y^{(i)}) x_j^{(i)}\end{aligned} \tag{6}$$

Here, we use AUROC as the indicator function.

$$AUC = \frac{\sum_{\mathrm{ins}_i \in \mathrm{positive\ class}} \mathrm{rank}_{\mathrm{ins}_i} - \frac{M \times (M+1)}{2}}{M \times N} \tag{7}$$

$\mathrm{rank}_{\mathrm{ins}_i}$ represents the serial number of the sample i (The probability score is ranked from small to large). M and N are the number of positive samples and negative samples. The $\sum_{\mathrm{ins}_i \in \text{positive class}}$ is to add the positive number. This is a discrete function so we cannot compute its derivation. In future work, we will discuss the use of other continuous functions as indicator functions. Finally, we get

$$\frac{\partial L}{\partial \theta_j} = \frac{w_i}{N} \sum_{i=1}^{N} (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)} \tag{8}$$

Figure 4 shows the curve of the category i weights as a function of the indicator. The the different colors of curves represent the different mean indicator scores of all classes except the category i. The abscissa represents the indicator score of the category i, and the ordinate indicates the weight w_i assigned to the category i. For each curve, with the increase of the indicator score of the category i, the learning of this category becomes sufficient, and the concentration on category i should be reduced. So the weight assigned to it is reduced slowly to avoid class imbalance and over-fitting. The curves of different colors shows that, when the average indicator scores of other categories increase, the weight assigned to category i also increases. This promotes the learning strength of the network model for category i.

3.3 Advantages

The design of the BAW has three advantages. (1) The algorithm calculates the weight by a certain number of samples, which avoid the problem that the calculation of a single sample is easy to fall into the noisy label. (2) The learning weight is calculated with the output of the network. This process can be carried out adaptively during training and reduces the tedious process of manual parameter adjustment. (3) The algorithm uses the evaluation indicator in

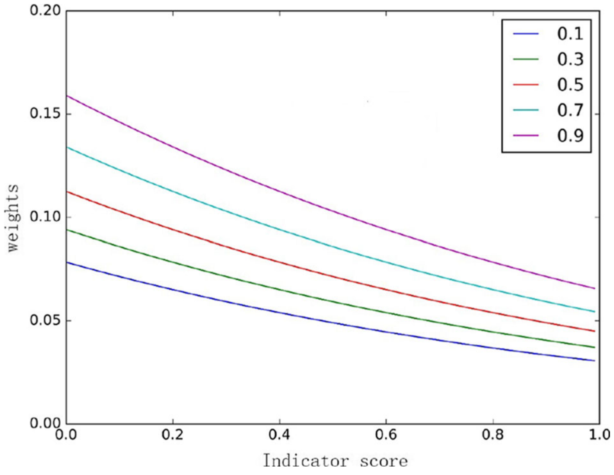


Fig. 4 Weights under different indicator scores. The the different colors of curves represent the different mean indicator scores of all classes except the category i

the training process to calculate class learning weight. This improves the performance of the final evaluation, even if these indicators are discontinuous and non-differentiable.

4 ChestNet

We designed an end-to-end chest disease recognition framework ChestNet, as shown in Fig. 1, based on deep learning modules, and applied it to the real medical dataset ChestX-ray14. Based on DenseNet121, ChestNet extract features from the perspective of channels and space through Squeeze-and-Excitation(SE) [14] and Spatial pooling [6]. Meanwhile, it use BAW loss to deal with the problems of class imbalance and noisy labels. On the task of pneumonia detection from chest X-rays, our ChestNet achieves the best results without any fancy tricks.

4.1 Network architecture

ChestNet uses DenseNet121 as the backbone to extract features. We express the input feature map of the l dense block as F_l, where $l = 1, 2, 3, 4$. The output can be expressed as $Dense(F_l)$. We then use the SE to weight the channels to get the input of $l+1$ Dense Block. It can be represented as

$$F_{l+1} = SE(Dense(F_l)). \tag{9}$$

The final output of DenseNet121 after extracting features, F_5, is convoluted to obtain a feature map with n number of channels. In the real experiments, $n = 14$, which represents 14 diseases. After, we use spatial pooling and sigmoid to get the final classification result.

4.2 Spatial pooling

Unlike global max pooling, which only cares about the areas with the highest response values. Spatial pooling hopes to learn the regions with both the highest and the lowest response in the feature map. Let C denote the number of channels in the feature map. U^C represents the feature map before pooling, and Z^C represents the mapping points after pooling. H_k is a series of pixel points from U^C. First average the k pixels with the largest response to get

$$z^{C+} = \max_{h \in H_{k^+}} \frac{1}{k^+} \sum H_{k,j} u^c(i, j). \tag{10}$$

Then average the k pixels with the lowest response to get

$$z^{C-} = \min_{h \in H_{k^-}} \frac{1}{k^-} \sum H_{k,j} u^c(i, j). \tag{11}$$

Finally, we get

$$Z^C = z^{C+} + \alpha(z^{C-}). \tag{12}$$

α is a hyperparameter used for adjusting the relative importance between z^{C+} and z^{C-}.

5 Experiment

5.1 Datasets

5.1.1 Simulation data

To establish an ideal experiment environment with controllable dataset, we simulate class imbalance and noisy labels on Cifar10. When simulating class imbalance, we randomly select 10% to 100% of samples from ten categories. The number of samples of our imbalanced 10 categories is 469, 1039, 1533, 1992, 2510, 3014, 3461, 3970, 4508, 5000. When generating noisy labels, we add 10%, 20%, 30%, 40% and 50% noise respectively to the original data. Noise includes Asymmetric Noise (AN) and Symmetric Noise (SN). SN randomly scrambles labels of different classes, and AN only scrambles labels of similar images. The noisy labels generated by AN can still provide some information. Therefore, SN is more challenging than AN. We combine Asymmetric noise/Symmetric noise and class balance/class imbalance in pairs, to get four experiment contexts.

5.1.2 Real data

The ChestX-ray14 dataset collected a total of 108,748 chest radiographs of 32,716 individual patients. Each image is labeled with one or more chest diseases. There are 14 common chest diseases in the whole dataset: Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax, Consolidation, Emphysema, Edema, Fibrosis, Pleural Thickening(PT) and Hernia. The number of samples varies from disease to disease, as shown in Table 4. The quantity of these categories is extremely uneven. The accuracy of the label is around 90%. Moreover, the noise is a mix of AN due to misdiagnosis and SN cause by text processing errors. To make an objective comparison with the previous methods, we use the official division of training, validation, and test list to divide the dataset into three parts.

5.2 Implementation

5.2.1 Simulation data

We perform fair tests on different loss functions in our four simulation data environments of Cifar10 data. The model uses ResNet18 [11] as the backbone, and uses SGD [1] with 0.9 momenta as optimizer. The weight decay is set to 5e-4, and the batch size is set to 128. The model is trained with 500 iterations and the learning rate is set to 0.1.

5.2.2 Real data

On the real medical dataset ChestX-ray14, we compare the classification quality of BAW with other loss functions on ChestNet. Moreover, to verify the robustness of these loss functions to noisy data, we manually add 2% and 5% noise to the ChestX-ray14 dataset. Besides, we also compare the results of our ChestNet with that of other methods. In the above experiments, we preprocess the data through cropping, horizontal flip, and normalization. We used SGD with 0.9 momenta as optimizer. The weight decay is set to 1e-4, and the batch size is set to 15. The model is trained with 20 iterations. The initial learning rate is set to 0.01, and will decay at the the 9th, the 12th and the 15th epoch with a decay rate of 0.1.

5.2.3 Other loss functions

In our experiments, we compare BAW loss with other four loss functions: Complement Cross Entropy (CCE) [16], Mean Absolute Error(MAE), Gradient Harmonizing Mechanism (GHM) [20] and Focal Loss (FL) [22]. The first three loss functions are described below:

CCE loss is formulated as:

$$H(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{[j]}^{(i)} \log \hat{y}_{[j]}^{(i)} \tag{13}$$

$$\tilde{c}(\hat{y}) = \frac{\gamma}{K-1}c(\hat{y}) \tag{14}$$

$$CCE = H(y,\hat{y}) + \tilde{c}(\hat{y}) \tag{15}$$

where N denotes the number of examples, K denotes the number of categories, $y^{(i)}$ denotes the true distribution, and $\hat{y}^{(i)}$ denotes the softmax multinomial prediction distribution. $c(\hat{y})$ denotes the cross entropy loss. γ should be tuned to decide the amount that complements the cross entropy, e.g., $\gamma = -1(\gamma < 0)$.

MAE loss is formulated as:

$$MAE = \frac{1}{K}\sum_{i=1}^{K} |y_i - \widehat{y}_i| \tag{16}$$

MAE calculates the average distance between the predicted value $\hat{y}_i$ and the real value y_i of the sample. It is worth mentioning that MAE is less sensitive to outliers and more inclusive than MSE. Because MAE calculates the absolute value of the error $y_i - \widehat{y}_i$, whether it is $y_i - \widehat{y}_i > 1$ or $y_i - \widehat{y}_i < 1$ without square term, the penalty is the same, and the weight is the same.

GHM loss is formulated as:

$$L_{CE}\left(p_i, p_i^*\right) = \begin{cases} -\log(p), \text{ if } p^* = 1 \\ -\log(1-p), \text{ if } p^* = 0 \end{cases} \tag{17}$$

$$GD(g) = \frac{1}{l_\in(g)}\sum_{k=1}^{N} \delta_\in(g_k, g) \tag{18}$$

$$\delta_\in(x,y) = \begin{cases} 1, & \text{if } y - \frac{\epsilon}{2} \le x < y + \frac{\epsilon}{2} \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

$$l_\in(g) = \min\left(g + \frac{\epsilon}{2}, 1\right) - \max\left(g - \frac{\epsilon}{2}, 0\right) \tag{20}$$

$$\beta_i = \frac{N}{GD(g_i)} \tag{21}$$

$$L_{GHM-C} = \frac{1}{N}\sum_{i=1}^{N} \beta_i L_{CE}(p_i, p_i^*) = \sum_{i=1}^{N} \frac{L_{CE}(p_i, p_i^*)}{GD(g_i)} \tag{22}$$

$L_{CE}\left(p_i, p_i^*\right)$ is denoted as binary cross entropy loss. The researchers proposed the gradient equalization mechanism. According to the proportion of the sample gradient modulus length distribution, a corresponding normalization is carried out, so that various types of samples have a more balanced contribution to the updating of model parameters, thus making the model training more efficient and reliable. Since gradient equalization essentially weights the gradients generated by different samples, and then changes their contribution, the weight added to the loss function can achieve the same effect, in GHM, the gradient equalization mechanism is realized by reconstructing the loss function. Gradient density is defined to describe the loss function. Following the physical definition of density (mass per unit volume), they define gradient density $GD(g)$ as the number of samples distributed in the area of unit value. Where g_k Represents the gradient of the kth sample. The gradient density harmonizing parameter is defined as β_i. Here, multiplied the sample number n is to ensure uniform distribution or only divide a unit area, the weight is 1, that is, loss is unchanged. It can be seen that the weight of samples with high gradient density will be reduced and the weight of samples with small density will increase. According to the GHM-C's calculation, the weights of simple negative samples and difficult abnormal samples will be reduced, so that the loss will be reduced, and the impact on model training will be greatly reduced. In order to improve the performance of the model, the weight of normal difficulty samples is increased, so that the model will focus more on the normal difficulty samples which is more effective.

5.3 Results

5.3.1 Simulation experiment

Table 1 show the results on balanced data (the upper part) and imbalanced data (the lower part) with AN. It is obvious that our method has better performance when the noise increases (for balanced data: AN>30%, for imbalanced data: AN>10%). When comparing the results of the upper and lower part, our method gets better performance on the class imbalanced data. It shows that BAW is robust to more serious noisy labels and class imbalance problems, and can be applied to complex datasets in practice. In Fig. 5, we show the change of test accuracy with the increase of epoch in different Asymmetric noise interventions in Cifar10. The corresponding results are shown in the second, third, fifth and seventh columns of the upper part of Table 1. We can find that test accuracy of different loss functions is very close when there is no noisy labels (or less noisy labels). As the ratio of noisy labels increases, the performance of the loss functions that is used to solve the problems of class imbalance

Table 1 Accuracy of different loss functions on Cifar10 with Asymmetric noise

Method	AN=0	AN=10	AN=20	AN=30	AN=40	AN= 50
CCE	92.46	87.52	82.14	79.81	71.88	68.05
MAE	83.62	58.33	56.46	55.42	57.83	51.76
GHM	**92.55**	87.07	83.02	79.24	76.57	66.83
FL	91.27	**88.11**	**84.23**	**82.41**	77.72	72.42
BAW	92.3	87.07	83.4	80.42	**77.89**	**75.42**
CCE	81.89	78.24	74.9	70.98	68.49	61.04
MAE	61.02	60.23	59.42	56.38	55.74	48.87
GHM	80.85	78.58	75.01	70.47	68.09	65.49
FL	82.08	**80.46**	75.27	70.71	68.64	65.03
BAW	**82.48**	79.86	**75.63**	**72.75**	**69.53**	**66.35**

The upper group is the results of class balanced data, the lower group is the results of class imbalanced data, and the bold figures represent the optimal result

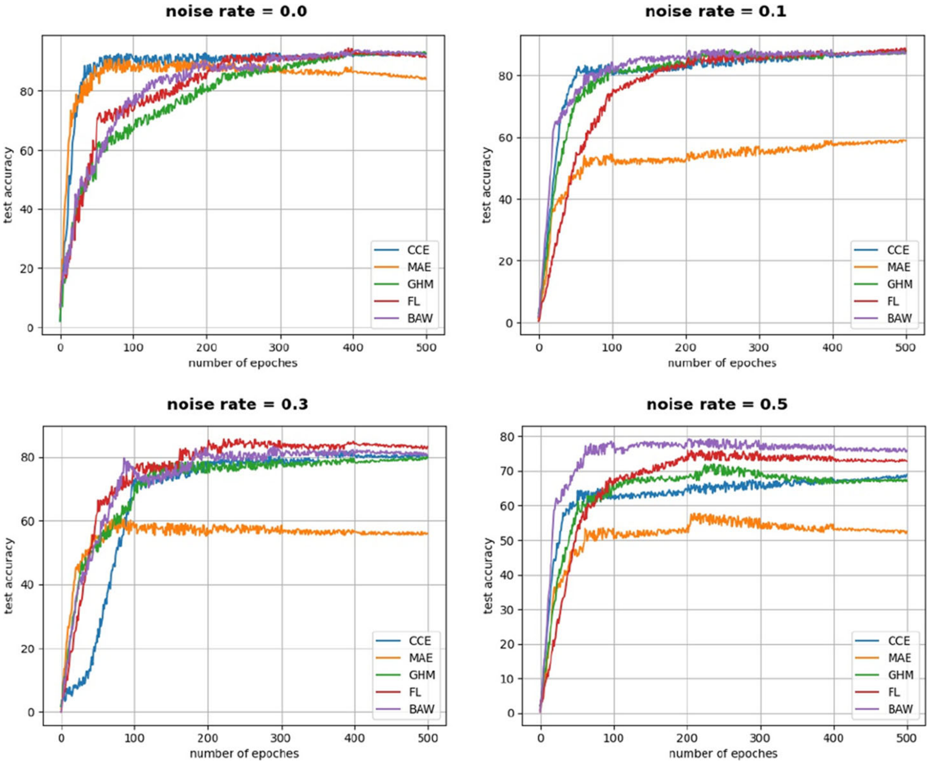


Fig. 5 This figure displays the change of test accuracy with the increase of epoch in different Asymmetric noise interventions in Cifar10. Different color represents different loss functions. The corresponding results are shown in the second, third, fifth and seventh columns of the upper part of Table 1

become worse, because they overfit the noisy data. In comparison, BAW is not sensitive to a large number of noise labels. When 50% of the labels are incorrect, BAW loss achieves better result that is 3% higher than the others. We further verify our method with SN which is more complex. BAW achieves best performance in almost all (i.e., 9/10) cases (Table 2).

5.3.2 Real experiment

We use the AUROC curve to measure model classification performance. The ROC curve takes false positive rate (FPR) as the abscissa and true positive rate (TPR) as the ordinate. The AUROC curve combines the probability of all thresholds to obtain an estimate of the classification performance.

We first compare the different loss functions on our proposed ChestNet. From the first column of Table 3, we can see that on real large-scale datasets without modification, BAW achieves best performance compared with other loss functions. As is shown in the second and third columns, the performance loss of BAW is smallest after noise addition, but the performance of other loss functions decreases significantly, which again proves that our method is more effective on dealing with complex, noisy labels and class imbalance problems in reality. The main reason for the ability difference is that, these loss functions do not take into account the existence of noisy labels when solving the problem of unbalanced data

Table 2 Accuracy of different loss functions on Cifar10 with Symmetric noise

Method	SN=0	SN=10	SN=20	SN=30	SN=40	SN=50
CCE	92.46	**85.87**	78.38	73.79	65.61	57.13
MAE	83.62	62.73	61.07	59.03	56.96	56.50
GHM	**92.55**	48.92	34.47	27.93	26.88	28.76
FL	91.27	83.4	78.42	66.45	62.74	57.94
BAW	92.3	83.93	**79.76**	**75.89**	**67.02**	**58.67**
CCE	81.89	63.83	59.24	55.1	51.21	47.89
MAE	61.03	56.80	56.10	50.98	50.03	46.77
GHM	80.85	35.66	27.64	25.76	21.08	18.85
FL	82.08	73.56	68	62.44	54.63	48.31
BAW	**82.48**	**74.15**	**69.57**	**64.03**	**57.50**	**49.19**

The upper group is the results of class balanced data, the lower group is the results of class imbalanced data. The best performance of loss function in each experiment group are highlighted

and long tail distribution. Therefore, when learning the distribution, the category with less data will be considered as marginal distribution, so these loss functions increase the weight for such data when training. For example, Focal Loss believes that the greater the difference between the model output and the label, the more difficult the sample is to learn, so the greater the weight should be. For a noisy label sample, the prediction and the ground truth ought to be quite different, so the network will over fit it under this idea. The most significant difference between BAW and these loss functions is that it does not pay attention to individual noisy label. It focus on the learning difficulty for a certain category by the AUROC indicators of known samples and calculate the corresponding weight of each category. Here we focus on two issues to better explain the working principle of BAW.

- What does the AUROC indicator represent? After a batch training, we calculate the AUROC on all known data. This means that data involved in the calculation have been fitted through network learning. So a high AUROC value means that the data of this category is well fitted, and vice versa. There may be two reasons for under fitting. The first is that the amount of data in this category is relatively small, so it is difficult to be fitted. The second is that although there are plenty of data, they are difficult to distinguish at the semantic level (It is worth noting that although there are noise labels in the training set, the impact on AUROC can be ignored due to the small number). At this time, the mechanism of AUROC is similar to that of Focal Loss. The difference is that Focal loss judges whether a specific sample is difficult to learn (increases the

Table 3 The results (AUROC) of different noise ratios under different loss functions in ChestX-ray14

Method	No artificial noise	2% artificial noise	5% artificial noise
CCE	0.8210	0.8086	0.8051
GHM	0.8156	0.8137	0.6869
FL	0.8113	0.7970	0.7822
BAW	**0.8279**	**0.8148**	**0.8116**

We highlight the highest score in each group. BAW performs best under all the different ratios weight of difficult samples, reduces the weight of easy samples). And our algorithm is to determine whether a category is difficult to learn (increases the weight of difficult categories, reduce the weight easy categories). So we can conclude that the use of AUROC as indicator in BAW can solve the problem of unbalance data distribution.

- What happens when a mislabeled sample passes through the network? First, its prediction score should be low on the given wrong label. However, because its weight is given by the AUROC of the wrong class, it will not get a huge weight compared with the weight given by Focal loss, thus avoiding network over-fitting on this sample. According to our observation, general label errors tend to occur in easy categories with large amount of data, so the given weight under this condition should be quite small. We conclude that BAW is not sensitive to noisy labels because it uses the AUROC indicator of known data as a buffer to alleviate the negative impact of the huge gradient caused by extreme samples.

Table 4 shows the detailed comparative experiments with other methods on each disease category. BAW achieves best result in most of the categories. In particular, our method significantly improves the performance on relatively small classes (i.e. Edema, Fibrosis, Hernia). Rajpurkar et al. [26] has discussed why his results are better than the others. We use the same backbone (DenseNet) in [26] as feature extractor and use the same data split scheme but different loss functions. CheXNet modified the loss function to optimize the sum of unweighted binary cross entropy losses. However, it can not surpass the performance of BAW in dealing with data imbalance, nor can it prevent the network from over fitting the noise label data.

It’s worth noting that the results of [19] in the third column perform better on Atelectasis, Effusion, Consolidation and Emphysema. However, one possible reason is that they did

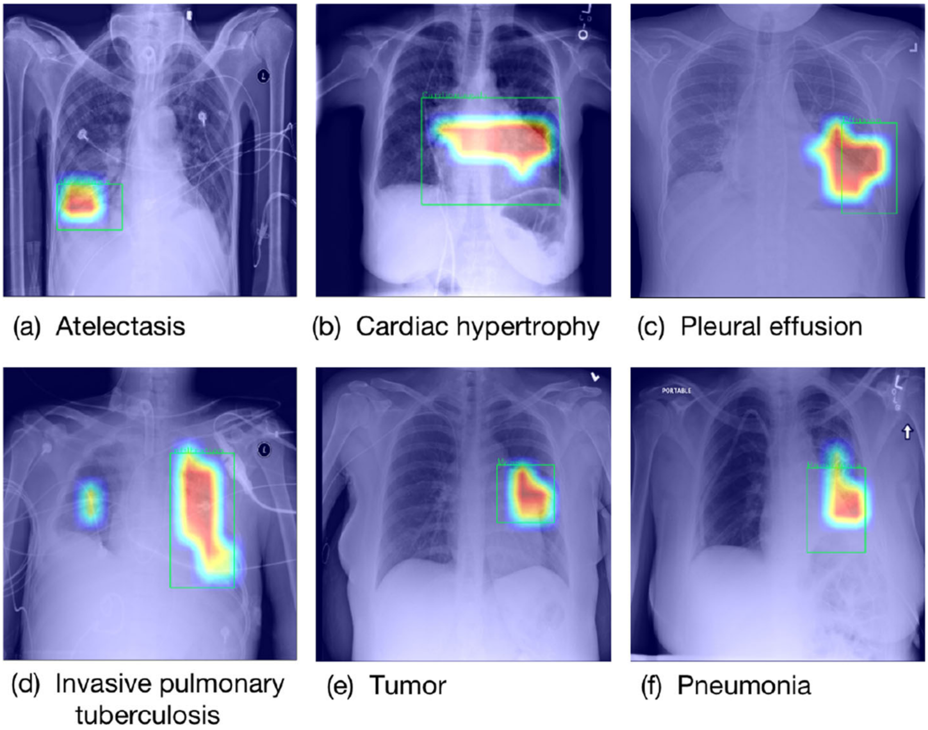
Table 4 The classification performance (AUROC) of our ChestNet and of other models on ChestX-ray14 for different diseases

Method	Wang [34]	Li [19]	DNetLoc [9]	CheXNet [26]	Ours
Official division	Y	N	Y	Y	Y
Atelectasis (8.15%)	0.7160	**0.8000**	0.7670	0.7795	0.7833
Cardiomegaly (1.96%)	0.8070	0.8700	0.8830	0.8816	**0.8941**
Effusion (9.41%)	0.7840	**0.8700**	0.8280	0.8268	0.8374
Infiltration (17.46%)	0.6090	0.7000	0.7090	0.6894	**0.7112**
Mass (4.06%)	0.7060	0.8300	0.8210	0.8307	**0.8462**
Nodule (4.47%)	0.6710	0.7500	0.7580	0.7814	**0.8047**
Pneumonia (1.65%)	0.6330	0.6700	0.7310	**0.7354**	0.7349
Pneumothorax (3.74%)	0.8060	0.8700	0.8460	0.8513	**0.8753**
Consolidation (3.30%)	0.7080	**0.8000**	0.7450	0.7542	0.7620
Emphysema (3.26%)	0.8350	**0.8800**	0.8350	0.8496	0.8489
Edema (1.77%)	0.8150	0.9100	0.8950	0.9249	**0.9360**
Fibrosis (1.19%)	0.7690	0.7800	0.8180	0.8219	**0.8262**
PT (5.28%)	0.7080	0.7900	0.7610	0.7925	**0.7965**
Hernia (0.16%)	0.7670	0.7700	0.8960	0.9323	**0.9339**
AVE	0.7381	0.8064	0.8066	0.8180	**0.8279**

The bold figures represent the optimal result not use the official dataset split. Guendel et al. [9] observed several limitations of the official split that training and test sets have different characteristics, which can be either label inconsistency or the fact that there are on average 3 times more images per patient in the test set compared with the training set. They computed several random splits, each of which leads to better performance and can increase AUROC from 0.8066 (the last figure in the forth column) to 0.841.

5.3.3 Visualization analysis

In this section, we will evaluate the effectiveness of the location positioning of the model in the chest disease detection task. We used the CAM method to build a thermal map for each disease category of each sample and visualize the most exciting area with the maximum response of the chest X-ray image, which is also the disease area predicted by the model. Figure 6 shows the comparison between the output by our model and the bounding box labeled given by doctors. Our network can preliminarily locate the disease area in a chest X-ray image by enlarging the response value of the target area in the feature map. The predicted disease area (red) is very close to the real lesion labeled area (green boundary box) from the perspective of visual evaluation. As shown in Fig. 6, our method expands the area with cardiac hypertrophy symptoms and provides more information for clinicians.



(a) Atelectasis

(b) Cardiac hypertrophy

(c) Pleural effusion

(d) Invasive pulmonary tuberculosis

(e) Tumor

(f) Pneumonia

Fig. 6 We use CAM [36] to visualize the attention of the network when doing classification. The green bounding box in the image is labeled by doctors to frame out the symptoms. From the perspective of visual evaluation, the disease area (red area) predicted by this model is very close to the real lesion labeled area (green bounding box)

Both sides of the chest X-ray with invasive pulmonary tuberculosis in the figure have this symptom. In contrast, each image in the dataset has only one disease boundary box, so the annotation ignored some boundary boxes of the same disease. Our approach gives a remedy so that we do not ignore any suspicious symptoms. Besides, the proposed model can still accurately identify and locate small target diseases such as masses and pneumonia. The final prediction of our model can provide potential candidate areas for further examination in clinical practice. Our model does not use any samples with boundary box labels during the training, which means that our model only uses the samples with category labels to achieve the preliminary positioning of chest diseases. This weakly supervised target detection diminishes the workload for labeling the disease boundary box. Simultaneously, the heat map generated by the model shows a promising ability to locate the disease area and can be widely used in clinical practice when lacking annotation information.

6 Conclusion

This paper proposes the BAW loss that solves the problem of class imbalance and avoids the over-fitting of noisy labels. Specifically, BAW uses the information of known samples to calculate weights for each category, and to guide the direction of network optimization for the next batch training. BAW is easy-to-implement and can be extended to various deep networks. First of all, this paper uses Cifar10 to simulate imbalanced classes and noisy labels. It is proved that BAW performs better with tough situation of noisy labels and class imbalance.

Besides, we also propose ChestNet, which is used to test our BAW loss in the large-scale medical dataset ChestX-ray14. The result shows that BAW is quite robust in real data.

Declarations

Conflict of Interests All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Bottou L (2012) Stochastic gradient descent tricks neural networks: tricks of the trade
2. Chawla N, Japkowic N, Kotcz A, Japkowicz N (2004) Editorial: special issues on learning from imbalanced data sets. Ann Nucl Energy 36(3):255–257
3. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2011) Smote: Synthetic minority over-sampling technique
4. Dan H, Lee K, Mazeika M (2019) Using pre-training can improve model robustness and uncertainty
5. Ding Y, Wang L, Fan D, Gong B (2018) A semi-supervised two-stage approach to learning from noisy labels. In: 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, pp 1215–1224
6. Durand T, Mordan T, Thome N, Cord M (2017) Wildcat: weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In: IEEE conference on computer vision pattern recognition
7. Fawcett T (2006) An introduction to ROC analysis[J]. Pattern recognition letters 27(8):861–874
8. Ghosh A, Kumar H, Sastry PS (2017) Robust loss functions under label noise for deep neural networks
9. Guendel S, Grbic S, Georgescu B, Zhou K, Comaniciu D (2018) Learning to recognize abnormalities in chest x-rays with location-aware dense networks. Springer, Cham

10. Haynes D, Corns S, Venayagamoorthy GK (2012) An exponential moving average algorithm. In: Evolutionary computation
11. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778. https://doi.org/10.1109/CVPR.2016.90
12. Huang G, Liu Z, Laurens V, Weinberger KQ (2016) Densely connected convolutional networks. IEEE Computer Society
13. Huang J, Qu L, Jia R, Zhao B (2019) O2u-net: a simple noisy label detection approach for deep neural networks. In: 2019 IEEE/CVF international conference on computer vision (ICCV), pp 3325–3333. https://doi.org/10.1109/ICCV.2019.00342
14. Hu J, Shen L, Sun G, Albanie S (2018) Squeeze-and-excitation networks[C]. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
15. Jindal I, Nokleby M, Pressel D, Chen X (2019) A nonlinear, noise-aware, quasi-clustering approach to learning deep cnns from noisy labels. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 64–72
16. Kim Y, Lee Y, Jeon M (2021) Imbalanced image classification with complement cross entropy
17. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto
18. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436
19. Li Z, Wang C, Han M, Xue Y, Wei W, Li LJ, Fei-fei L (2017) Thoracic disease identification and localization with limited supervision
20. Li B, Liu Y, Wang X (2018) Gradient harmonized single-stage detector
21. Lin TY, Maire M, Belongie S, Hays J, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision
22. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. IEEE Transactions on Pattern Analysis Machine Intelligence PP(99):2999–3007
23. Liu XY (2006) Exploratory under-sampling for class-imbalance learning. In: International conference on data mining
24. Northcutt CG, Jiang L, Chuang IL (2021) Confident learning: estimating uncertainty in dataset labels. J Artif Intell Res (JAIR) 70:1373–1411
25. Patrini G, Rozza A, Menon AK, Nock R, Qu L (2017) Making deep neural networks robust to label noise: a loss correction approach. In: IEEE conference on computer vision pattern recognition
26. Rajpurkar P, Irvin J, Zhu K, Yang B, Mehta H, Duan T, Ding D, Bagul A, Langlotz C, Shpanskaya K (2017) Chexnet: radiologist-level pneumonia detection on chest x-rays with deep learning
27. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. IEEE
28. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge[J]. International journal of computer vision 115(3):211-252
29. Szegedy C, Wei L, Jia Y, Sermanet P, Rabinovich A (2014) Going deeper with convolutions. IEEE Computer Society
30. Tanaka D, Ikami D, Yamasaki T, Aizawa K (2018) Joint optimization framework for learning with noisy labels. IEEE
31. Vahdat A (2017) Toward robustness against label noise in training deep discriminative neural networks
32. Voets M, Mllersen K, Bongo LA (2018) Replication study: development and validation of deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. PLos ONE
33. Wallace BC, Small K, Brodley CE, Trikalinos TA (2012) Class imbalance, redux. In: IEEE international conference on data mining
34. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM (2017) Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. IEEE
35. Yao J, Wang J, Tsang IW, Zhang Y, Sun J, Zhang C, Zhang R (2018) Deep learning from noisy image labels with quality embedding. IEEE Trans Image Process 28(4):1909–1922
36. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: CVPR
37. Zhou B, Lapedriza A, Khosla A, Oliva A, Torralba A (2017) Places: A 10 million image database for scene recognition[J]. IEEE transactions on pattern analysis and machine intelligence 40(6):1452–1464

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.