UNIVERSITÀ
DEGLI STUDI
FIRENZE

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione
Curriculum: Ingegneria Informatica

Meta-advisor learning for dataset annotations issues in the image classification task

Candidate
Simone Ricci

Supervisors
Prof. Alberto Del Bimbo

Prof. Marco Bertini

PhD Tiberio Uricchio

PhD Coordinator
Prof. Fabio Schoen

ciclo XXXIV, 2018-2022

Università degli Studi di Firenze, Dipartimento di Ingegneria
dell'Informazione (DINFO).

Alla mia famiglia e amici

Abstract

Automatically classifying images is a long-standing problem in computer vision. While recently Convolutional Neural Networks (CNNs) have become the state-of-the-art approach and have made a substantial improvement in accuracy when enough supervision is available, they still do not solve the task entirely. Many real-world applications demand machines capable of classifying the vast quantity of object classes known, even when the supervision is limited or unavailable. Deep learning models require huge labeled data collections for training. Creating these datasets has a high cost and requires a very long time as each sample needs to be manually annotated by experts of the domain. The Internet has made accessible a massive collection of images annotated by ordinary users or associated with text and other metadata that may suggest their content. Given their context, these data have mislabeled samples that could induce performance reduction or bad behavior in the trained model. In addition, these collections may contain an imbalance in the distribution of the number of samples of each category. The frequency of objects in the real world often shows a long-tailed distribution where few classes dominate. Extremely specific concepts have low availability of samples despite the large amount of online data. Training CNNs on these data fail to classify the underrepresented tail classes. The challenge is to use such zero-cost images while employing minimal human effort in labeling. It is thus mandatory to look for new ways of handling noise and long-tail distributions, designing new image models and strategies, that may work even with few correctly labeled images.

In this thesis, we developed the novel concept of advisor networks to address both the noisy label problem and the long-tail distribution problem. This network helps the classifier by taking advantage of two new methods that exploit information extracted from it.

In the first part of the thesis, we propose a meta-learned attention ap-

proach that lets the classifier focus only on the meaningful part of the visual feature of an image, according to the advisor network. That gives the classifier the ability to take advantage of examples with noisy annotation, improving the model generalization. We show that meta attention is an efficient approach to handle synthetic and real-world noise for the classification task.

In the last part of the thesis, we apply the previously meta-learned attention approach to the long-tail distribution problem for image classification. We demonstrate that the method is an effective solution to handle this type of training data issue. We also introduce a new meta-activation feature suited for the class imbalance problem. Through this, the network of advisors learns to avoid the discouraging gradients of common classes that harm the proper learning of rare ones. We show the effectiveness of this meta-activation even on the problem of noisy labels. Our two methods can be used jointly by operating on different sections of the classifier. Their cooperation allows for greater efficacy of the advisor network in helping the classifier. We introduce a new dataset setting, where the noisy labels problem and the long-tail distribution are present conjunctly, to prove the adaptability of our solution.

Keywords: *machine learning, deep learning, meta-learning, noisy labels, long-tail, attention, discouraging gradient, computer vision.*

Publications

International Journals

Submitted

1. **Simone Ricci**, Tiberio Uricchio and Alberto Del Bimbo. "Meta-learning advisor networks for long-tail and noisy labels in social image classification", *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2022. (Submitted after major revision)

International Conferences and Workshops

1. **Simone Ricci**, Tiberio Uricchio and Alberto Del Bimbo. "Learning advisor networks for noisy image classification", in *21st International Conference on Image Analysis and Processing (ICIAP)*, Lecce (Italy), 2021, flore id 2158/1253562

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural intelligent systems learn new concepts in a world containing a myriad of information that is not always correct and can remember items seen even a few times. During their education, humans can focus more on only the proper concepts and be able to exploit even those data that have a wrong association. In addition, many notions that have been learned quickly or studied a few times are remembered even though they are overwhelmed by other prevalent information. Most supervised learning systems work under the assumption that all training data is correct and perfectly balanced. In contrast to humans, they suffer when many data have a wrong association or when a few concepts dominate over all other categories. Moreover, modern classification systems require a large amount of data to achieve optimal performance. Labeling examples is an expensive and time-consuming process that requires the participation of expert annotators. Online sources offer a huge quantity of images annotated by everyday users and provided often with text and other metadata that may indicate their content. Due to their nature, these generated datasets can contain mislabeled samples [1]. In a real-world scenario, collecting a sufficiently large number of representative examples for each category is not always possible. Instances of some concepts are difficult to find, or their availability is limited because they are poorly documented. Object frequency in the natural world often exhibits a long-tailed distribution where a tiny number of categories dominate the others [24]. This phenomenon leads to a severe imbalance within the training dataset that causes classification systems to over-learn the predominant classes and under-represent the tail ones. The problems of noisy annotations

and long-tail distribution are actual and seriously affect the efficiency of the most modern classification models [10,25]. Often the noise is not predictable in advance, and the distribution of concepts varies according to the specific topic and task. The solutions proposed in the literature typically require a well-structured type of annotation noise, or the frequency of the categories, addressing each of these aspects individually.

In this thesis, we introduce the new concept of advisor network that can help the classifier address both noisy labels and long-tail problems separately and jointly. In the following sections, we discuss, confront and examine relevant literature providing additional technical details of this dissertation. Finally, in the last section, we introduce the contributions made in this thesis.

1.1 Noisy Annotation Learning

The problem of noisy annotation has been extensively studied in literature [17,28,35,45,48,50,57]. Modern classification systems, and all machine learning methods in general, are prone to performance degradation when noise is present in the training label. Moreover, convolutional neural network (CNN) architectures, trained with stochastic gradient descent (SGD) optimizers, can fit large training datasets composed entirely of random labels [75]. As demonstrated by [3] when a training dataset has mostly correct annotations, deep neural models do not memorize the data, but they learn the predominant patterns shared among the training samples only in the initial part of the learning. For [3], this behavior happens thanks to explicit regularization strategies and the design of deep neural networks, which inherent distributed and hierarchical representation. The empirical results of [56] demonstrate a strong learning tendency of deep CNNs, in contrast to memorization. However, other studies have reported somewhat contradictory results to those just mentioned. Label noise can have a significant impact on the accuracy of CNN when applied to face recognition tasks [65]. Training on a smaller dataset with verified annotations is better than training on a much larger dataset with significant label noise. The trade-off between memorization and learning, as suggested by [3], depends on the nature and richness of the data, amount of label noise, model architecture, as well as training procedures along with regularization. As shown by [42], label noise conditions the local intrinsic dimensionality of the features, learned by a deep learn-

ing model. Local intrinsic dimensionality [22] measures the dimensionality of the underlying data manifold. Given a data point x_i, it quantifies the rate of encountering other data points as the radius of a ball centered at x_i grows. During the training on data with noisy labels, when the model learns the dominant patterns in the data, the local intrinsic dimensionality of the features initially decreases. Instead, the dimensionality starts to increase when the model begins to overfit on samples with incorrect labels [42].

1.1.1 Label cleaning

Identifying and either fixing or discarding noisy label data samples are the purposes of methods belonging to this category. The approach introduced by [64] made use of two CNNs trained in parallel on two different datasets, where the first was small and verified, instead the second was large and with noisy annotations. With the verified dataset, a CNN learned to clean the noisy annotations. The other CNN was optimized to solve the main classification task with the cleaned noisy data. Experiments demonstrated that this approach was more effective than training on the noisy dataset followed by a fine-tune on the verified dataset. The method CleanNet, proposed by [34], compared the feature vector of a noisy label sample with the representative feature vector of its class calculated from a small clean dataset. The similarity between these vectors determined whether the label is correct. For image classification, CleanNet assigned weights to training samples based on the similarity. An improved version of CleanNet was introduced by [20]. They estimated the correct labels with an iterative framework removing the need for a verified dataset. Moreover, multiple prototypes feature vectors represented each class, instead of the only one used in [34].

1.1.2 Noise layer

A well-studied strategy is adding a "noise layer" to the end of deep neural networks. The noise layer, introduced by [58], corresponds to multiplication with the transition matrix between noisy and corrected labels. It is learned in parallel with the network weights using error back-propagation. Instead, an Expectation-Maximization optimizing method, developed by [17], estimated the parameters of the noise layer. Further, they applied their model to the case where the label noise also depends on image features. An estimate of correct labels and annotator confusion matrices was introduced by [62] to

address the case of noisy labels obtained from multiple annotators. With the ambiguity in the concurrent estimation of correct labels and annotator confusion matrices, the traces of these matrices had to be penalized. Instead, for image classification [44] presented two CNNs, jointly trained, to separate the object presence and relevance. An extra latent variable was introduced to the previous strategy to measure the reliability of noisy labels [72].

1.1.3 Loss functions

In the literature, a large number of methods, instead of changing the network architecture or training data, keep these intact and make modifications to the cost function. The conditions for the robustness of a loss function were well-studied in [16]. In [47] a "B-correction" and "F-correction" were presented to improve the robustness of the model loss function. These two methods take advantage of an error confusion matrix T, defined as $T_{i,j} = p(\tilde{y} = e^j | y = e^i)$ where y and $\tilde{y}$ are the correct and noisy labels respectively. When T is non-singular, "B-correction" strategies was $l_{corr}(\circ p(y|x)) = T^{-1} l(\circ p(y|x))$. A linear weighting of the loss values for each category was applied, where the weights are the probability of the correct label given the observed label. A small verified dataset with clean labels was used by [21] to estimate T.

1.1.4 Data re-weighting

The methods in this category aim to reduce the weight of training samples that are most likely to have incorrect labels. A meta-learning approach [52] weigh each training data. Weights were optimized by minimizing the loss on a verified auxiliary dataset with clean annotations. This weighting scheme was equivalent to assigning larger weights to training samples similar to the clean auxiliary data in respect of both the learned features and optimization gradient directions. Rather than adopting a pre-defined weighting scheme, a multi-layer perceptron model MW-Net [57] composed of a single hidden layer was used to learn a proper weighting strategy for a specific task and dataset. This method also needs a small dataset with clean labels. The learned weighting scheme on datasets with noisy labels was similar to those proposed in other studies. MW-Net learned to down-weight samples when noisy labels were present.

1.1.5 Training procedures

Many strategies that exploit curriculum learning [4] have been proposed to address the label noise problem. Curriculum learning is a training process where samples are presented to the model in order of difficulty or complexity. Mentor-Net [26] was an LSTM network that generates weights on the training samples to provide a curriculum to a second network, called Student-Net. A knowledge distillation approach was adopted by [37]. An auxiliary model, optimized on a small dataset with clean labels, guided the training of the main model through pseudo-labels generated as a convex combination of the noisy label and the label predicted by the auxiliary model. Based on the label transition matrix, they introduced a knowledge graph to reduce the risk of the auxiliary model overfitting on the small clean dataset. A convex combination of labels predicted by the model at its current training stage and noisy labels was used by [50]. During learning, the model accuracy increased, and its predictions could be weighted more strongly, so gradually forgetting the original erroneous labels. Co-teaching [18] further developed this idea, whereby the two networks identified mislabeled samples and shared the updated information with the other network. This method also improves computational efficiency since unnecessary updates are avoided on easy data samples once the models predict the correct label on those samples. Similar to [18], a meta-learning objective was introduced by [36] to encourage consistent predictions between a student model. Mixup [76] generated new training data and labels through a convex combination of pairs of them. Given two randomly selected training data and label pairs (x_i, y_i) and (x_j, y_j), a new pair was generated as $\tilde{x} = \lambda x_i + (1 - \lambda)x_j$ and $\tilde{y} = \lambda y_i + (1 - \lambda)y_j$, where $\lambda \in [0, 1]$ is sampled from a beta distribution. Mixup is primarily a data augmentation and regularization strategy with a remarkably efficacious for combatting label noise.

1.2 Long-Tail Learning

Deep long-tail learning aims to train a deep neural network model from a collection of data that follows a long-tail class distribution. A small fraction of classes has a dominant number of samples concerning the other ones associated with only a few data. The models are usually evaluated to perform well on all categories, i.e. on a clean and balanced test set. However, the class imbalance of training sample numbers causes deep network models training

to be very challenging. Imbalanced data across classes make deep models overly fit the dominant classes and fail in the underrepresented tail classes. Moreover, the lack of tail-class samples makes classification hard on those concepts.

1.2.1 Class Re-balancing

Class re-balancing, a mainstream paradigm in long-tailed learning, seeks to balance the training sample numbers of different classes during model training. The class re-balancing paradigm aims to balance the number of training samples of different classes during model optimization. In the last few decades, one of the widely-used methods to resolve the class imbalance is re-sampling [6, 15, 19, 39]. Over-sampling and random under-sampling were the first developed strategies belonging to this category. Over-sampling randomly repeats the samples from tail classes until the class balance is achieved, while under-sampling randomly discards the samples from head classes. These two methods can't address the problem completely. Over-sampling tends to overfit tail classes when the imbalance ratio between the head and the tail class is very high. Instead, under-sampling tends to degrade the model performance on head classes. Recent long-tail learning studies develop different sampling methods, including class-balanced re-sampling. The basic strategy is instance-balanced sampling, where each data has an equal probability of being selected. Instead, in class-balanced sampling, each class has an equal possibility of being selected. An alternative of instance-balanced sampling is square-root sampling [43], where the probability for each class to be sampled is related to the square root of example number in the corresponding class.

1.2.2 Cost-sensitive Learning

The cost-sensitive learning branch aims to re-balance with loss values adjustments specific for each class during training [14, 59, 79]. The two main categories that fall under this approach are class-level re-weighting and class-level re-margining. For the first one, the most intuitive method was weighted softmax loss, which directly exploits label frequencies for loss re-weighting. Also, Balanced Softmax [51] uses the label frequencies to alleviate the bias of class imbalance by prior knowledge. They demonstrated how their loss minimizes the generalization bound for multiclass Softmax regression. Class-balanced loss introduced the concept of effective number to estimate the

expected number of samples from different classes. It is an exponential function of the number of training samples, instead of label frequencies. The class-balanced loss imposes a class-balanced re-weighting term inversely proportional to the number of classes. Long-tail class distribution usually raises the prediction hardness of tail classes because their prediction probabilities are lowered from head classes. Focal loss [38] used the prediction probabilities to inversely re-weight classes. It assigned higher weights to the harder tail classes and lower weights to the easier head classes. The increase in prediction hardness of the tail classes is due to an over-suppression of the negative gradient coming from the dominant categories [60]. In softmax or sigmoid cross-entropy, a positive sample for one class can be seen as negative for the others. The tail classes receive more negative gradients than positive ones. Equalization loss [60] introduced a weighting term in the loss, based on class frequency, to avoid discouraging gradients on the tail-class samples. Meta-Weight-Net [57], driven by a balanced validation set, exploited a loss weighting function learned by a one-layer MLP to let the main model fit the long-tail distribution. The class weights were learned from verified data as for the noisy label problem 1.1.4. Unlike the re-weighting approach, class-level re-margining aims to adjust the minimal margin (i.e., distance), between the learned features and the model classifier, for different classes. Expanding the soft margin loss [31,66], label-distribution-aware margin (LDAM) [5] enforced class-dependent margins based on label frequencies and facilitated tail classes to have larger margins. Their method was not empirically acceptable to solve the class imbalance problem. Thus, LDAM introduced a deferred re-balancing optimization schedule.

1.2.3 Prototype learning

Learning class-specific feature prototypes is the objective of prototype learning-based methods developed to improve long-tailed learning performance. The first to explore this idea was Open long-tailed recognition (OLTR) [40]. A visual meta-memory containing discriminative feature prototypes was used to augment the original features of the training data. Their test set includes the head, tail, and open classes, where open indicate categories that do not exist in the training set. Moreover, a self-attention mechanism was applied in OLTR to enhance feature learning.

1.2.4 Decoupled Training

Decoupled training split the learning process into representation learning and classifier training. [27] introduced this two-stage training scheme for the long-tail problem. They considered different sampling approaches for representation learning and then evaluated various classifier training schemes, fixing the previously trained feature extractor. They tested four methods to re-train the classifier: the class-balanced sampling, the τ-normalized and the nearest class mean classifier, and a trainable weight scaling approach. According to [27], instance-balanced sampling was the best strategy for representation learning.

1.3 Contributions

In this thesis, we focus initially on the noisy label learning classification problem, where the image annotations are not always correct and verified. We were interested in finding a way to take advantage of all the visual information coming from the training set, including the images with erroneous annotation. First, we proposed a meta-attention approach, that mimics the human attention, automatically learned through a meta-learned advisor network, developed to help the main classifier address the noisy label problem. Our method could be leveraged for different types and levels of annotation noise. In particular, we were interested in real-world noise typically found in collections of images gathered from online resources.

The online collected training datasets also suffer from the long-tail class distribution problem. We developed a new meta-activation function, learned from an advisor network, to wisely avoid the huge amount of discouraging gradients that suppress proper learning in rare classes. The strategy of weighting discouraging gradients can also be applied to other problems besides imbalance, such as noisy labels. Because our activation function is trained through a verified and balanced auxiliary dataset, it could easily adapt to the noisy label problem. The same reasoning has been applied for the meta-attention method, applying it to the case of long-tail distributions. Our two techniques, affecting different aspects of the classifier, were used jointly to address the two, even concurrent, problems typical of online collected training datasets.

In summary, we show several contributions in this dissertation:

Learning advisor networks for noisy label image classification

In Chapter 2 we addressed noisy annotation learning using a novel advisor network that helps the main model through a meta-attention method. Our method applied an automatically learned attention mask to the classifier's visual features. In this way, the classifier focused only on the meaningful information and could use parts of the mislabeled visual data to improve performance on the category assigned to the image. While weighting loss strategies aim to remove the influence of noisy samples, our method helps the classifier without altering the loss value of any data. We achieve state-of-the-art results on synthetically generated datasets with heavy noise levels, and most importantly on a dataset containing real-world noise.

Meta-learning advisor networks for long-tail and noisy labels image classification

In Chapter 3 we present a meta-activation function that avoids discouraging gradients of dominant concepts to solve the long-tail learning problem. We designed a new advisor network that learned a weight for each class to be applied in the activation function. Proved the efficiency of this new approach on standard evaluation dataset for long-tail distribution, we extended this method to the noisy label problem. We achieved good results on synthetically generated and real-world noisy datasets. This showed the applicability of discouraging gradients elimination to a different task than the long-tail class problem. We studied the application of our meta-attention method to solve the unbalance issue, obtaining great results on standard evaluation datasets. Since our approaches involve different classifier properties, we decided to apply them jointly, getting the state-of-the-art result on the real-world noise dataset. We finally introduced a new synthetic dataset containing noisy and long-tail distribution annotations to demonstrate the adaptability of our solution in handling multiple dataset issues.

Finally, in Chapter 4, conclusions of the dissertation and future challenges of advisor network related to online generated datasets are discussed.

Chapter 2

Learning advisor networks for noisy label image classification

In this paper, we introduced the novel concept of advisory network concept to address the problem of noisy labels in image classification. Deep neural networks (DNN) are prone to performance reduction and overfitting problems on noisy training data. Weighting loss methods aim to mitigate the influence of noisy examples during the training, completely removing their contribution. This prevents DNNs from learning wrong associations between images and their correct labels but reduces the amount of data used, especially when most of the examples are noisy. Differently, our method weighs the feature extracted directly from the classifier without altering the loss value of each data. The advisor helped to focus only on some part of the information present in noisy examples, allowing the classifier to leverage that data as well. We trained it with a meta-learning strategy so that it can adapt throughout the training of the main model. We tested our method on CIFAR10 and CIFAR100 with synthetic noise, and on Clothing1M that contains real-world noise, reporting state-of-the-art results. [1]

[1]The part of this chapter has been published as "Learning advisor networks for noisy label image classification" at the *21st International Conference on Image Analysis and Processing (ICIAP), 2021* [54].

Figure 2.1: Clothing images obtained from online shopping websites. Annotations are generated from the description of the clothes delivered by the vendors. The red crossed images are mislabeled, instead the green ones are correct.

2.1 Introduction

Modern image classification systems are based on using deep neural networks models that are trained on a huge number of labeled images [32]. Due to the extreme cost of labeling such an amount of images and difficulty in covering many concepts, researchers recently have looked into methods that generate labels automatically. One significant line of research exploits available labeled images from non-experts (e.g. from social networks, online stores) that can be easily retrieved in large quantities but may have mislabeled [1]. Figure 2.1 shows some examples of these data.

Deep neural networks typically consist of a large number of parameters that are highly shared among feature dimensions and states, enabling flexibility in learning different tasks and classes. This flexibility has the advantage to lead to strong discriminative models unless data is corrupted by noise, leading to performance reduction and overfitting problems [25]. Recent methods tried to address the problem by using curriculum learning [4], directly estimating the noise in the set [21], or measuring the confidence of the network during training [33], also using another co-trained network [18]. The idea was usually to understand samples out of distribution and reduce their influence on the learning by dampening their loss or decreasing their impact directly from the training set.

We proposed a meta-learning approach to address the problem of noisy labels in image classification based on an advisor network, developed to help

the classifier. While a standard image classification model is trained, the advisor network observes the main network activations and adjusts features at training time when noisy label images are identified as input. This allows the classifier model to get information even from mislabeled samples where some noise structure is present. We only retained the main model as the final classifier, while the advisor was discarded. Unlike the teacher-student paradigm, the advisor network was not trained to solve the image classification task, but only to help the learning process of the classifier model by its altering activations.

2.2 Related Works

In literature, the research on the noisy label learning problem was very active since machine learning systems are prone to performance degradation when noise is present in the training label [45, 48]. Loss correction was a well-treated technique to mitigate the effect of noisy samples on the classifier network. Works like Reed [50], F-correction [17], GLC [21], M-correction [2] and S-adaptation [47] made loss adjustments based on the estimated corruption probabilities matrix, changing the wrong labels to the correct ones. In [42, 61, 73] the noise distribution was modeled by linearly combining the noisy label with the output of the network. Re-weighting approaches assigned a weight to each data instance, avoiding the contribution of a noisy sample to the training by giving it a lower weight value. MentorNet [26] and MentorMix [25] found the latent weights through data-driven curriculum learning and student-teacher paradigm respectively. Some works used augmentation strategies that encourage the main model to behave linearly in-between training examples like Mixup [76], Advaug [7]. DivideMix [35] dynamically separated training data into clean and noisy sets to optimize two diverged networks with a semi-supervised strategy. In contrast, our method took advantage of an advisor network that alters activations of the main classifier to increase its performance, without isolating noisy label samples from the clean ones.

The noisy labels problem was also addressed by [36, 52, 68] with a meta-learning approach. These methods leveraged a small clean validation dataset to implement the meta-learning scheme. For example, L2R [52] weighed each sample by giving less importance to the noisy one. MLNT [36] imitated regular training with fabricated noisy labels. MLC [68] estimated the corruption

probabilities matrix to adjust the training loss values. MW-Net [57] automatically determined an explicit weighting function that can easily fit the biased training data, and it worked on both the noise and imbalance training problems. Differently from them, our advisor network modifies the network activations using a meta-attention layer to increase the performance of the main classifier during its learning, despite the noise in the training labels.

2.3 Contributions

In summary, our contribution are:

- We propose the novel concept of advisor network, i.e. the use of an additional network only at training time, learned by meta-learning. The advisor helps the main classifier to address noisy label learning problems.
- We introduce a Meta Feature Re-Weighting (MFRW) method that automatically generates an attention mask on the visual features of the classifier so that it focuses only on the information in an image deemed relevant by the advisor network.
- We test our approach in presence of artificial noise and on a popular real-world noisy dataset, obtaining state-of-the-art performance on Clothing1M.

2.4 Proposed Method

We developed a method that can handle images with noisy labels during the training of deep neural networks for the image classification task. Our idea originates from the idea that even an example associated with an erroneous annotation can contain information that can contribute to a greater generalization of the network. The model should focus on only a few convenient parts of this data. Our approach has been to exploit an attention mechanism to enhance the useful portions of the visual information and lower the rest. We developed an auxiliary advisor network that automatically learned a function that weighs the visual features extracted from a DNN during its training. This advisor network was aware of the state of the main model thanks to a meta-learning optimization strategy (Figure 2.2). We introduced

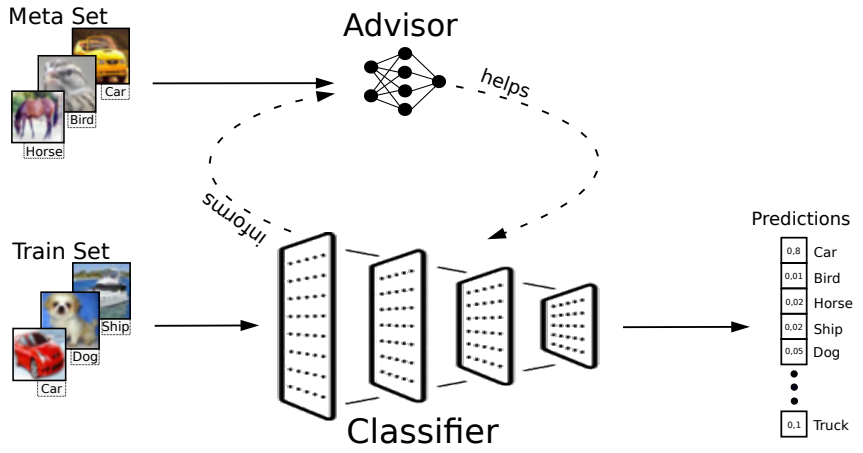

Figure 2.2: General overview of our system. An advisor network assists an image classifier by exploiting an auxiliary meta-set to reduce the effects of noisy annotations while learning on the training set.

a new method Meta Feature Re-Weighting (MFRW) which acts like a meta-attention layer. Different from weighting loss methods that tend to remove the influence of noisy examples during the training, our MFRW can take advantage of them.

In Section 2.4.1, we introduce meta-learning basics and formulation typical of methods that learn robust deep neural networks from noisy labels. Then in Section 2.4.2, we explain our method showing the architecture of the whole process. Finally, the learning process of the classifier together with the advisor network is described in Section 2.4.4.

2.4.1 Background meta-learning

In general, meta-learning (ML) refers to the process of improving a learning algorithm over multiple learning episodes, it is also called learning to learn. ML is divided into two algorithms: an inner (or base) and an outer (or upper/meta) algorithm. The inner one solves the main task minimizing an objective function, for image classification we have a convolutional neural network and the cross-entropy loss respectively. Instead, the outer algorithm

updates the inner one such that it improves also on an outer objective function. When the objective functions are the same for both algorithms, the outer algorithm can help the inner one to work well on new data distribution. If the new distribution is a smaller version of training data but free of errors and balanced, it is possible to train the outer algorithm to solve the problem of noisy or imbalanced labels inside the main training data. We refer to this distribution as meta-set. As in [57], the outer algorithm can be a multilayer perceptron network, called meta-model, that learns automatically how to address these problems helping the main image classifier during its learning. We introduce the symbols useful for understanding ML in this particular setting and how the entire learning process is divided, describing the [57] algorithm for simplicity.

Let $D^{train} = \{x_i^{tra}, y_i^{tra}\}_{i=1}^{N}$ be the training set with noisy or imbalanced annotations, where N is the total number of samples, composed of an image x_i and the correspondent one-hot label y_i over C classes. The main DNN model is defined as $\Phi(\cdot; w)$, where w are its parameters. The prediction on an input image x is $\hat{y} = \Phi(x; w)$ and the optimal parameters w^* are obtained by minimizing the softmax cross-entropy loss $\ell(\hat{y}, y)$ on the training set. Let $D^{meta} = \{x_j^{meta}, y_j^{meta}\}_{j=1}^{M}$ be the meta-set, a well verified and balanced version of training one but much smaller, $M \ll N$. The meta-model is defined with $\Psi(\cdot; \theta)$, parameterized by θ. In [57] the optimal parameter w^* is derived using the following weighted loss:

$$w^*(\theta) = \underset{w}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{V}_i^{tra}(\theta) \mathcal{L}_i^{tra}(w) \tag{2.1}$$

where $\mathcal{L}_i^{tra}(w)$ is the loss value and $\mathcal{V}_i^{tra}(\theta) = \Psi(\mathcal{L}_i^{tra}(w); \theta)$ is the weight predicted by the meta-model for the i-th training example. The meta-model is trained minimizing the loss of previously updated $\Phi(\cdot; w^*(\theta))$ on the meta dataset D^{meta}:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_j^{meta}(w^*(\theta)) \tag{2.2}$$

where $\mathcal{L}_j^{meta}(w^*(\theta)) = \ell(\Phi(x_j^{meta}; w^*(\theta), y_j^{meta}))$ is the softmax cross-entropy loss for the j-th meta example.

Both equations Eq. (2.1) and Eq. (2.2) can be solved by alternating optimization through gradient descent. An online strategy, that is divided into three main steps, can be adopted to update θ and w through a single optimization loop. This guarantees the efficiency of the algorithm and its conver-

gence [57]. In the first step, called Virtual-Train, the original DNN will not be updated and the optimization is carried out on a virtual model that is the clone of the original one. We consider the t-th iteration and the related mini batches $\mathcal{B}^{train} = \{(x_i^{tra}, y_i^{tra})\}_{i=1}^{n}$ and $\mathcal{B}^{meta} = \{(x_j^{meta}, y_j^{meta})\}_{j=1}^{m}$, where n and m are the size of mini-batch respectively. The virtual update is derived by:

$$\hat{w}(\theta) = w - \alpha \frac{1}{n} \sum_{i=1}^{n} \mathcal{V}_i^{tra}(\theta) \nabla_w \mathcal{L}_i^{tra}(w) \tag{2.3}$$

where w are its parameters at the current iteration and α is the learning rate for the DNN. Keeping in mind the virtual update previously carried out, in the successive step called Meta-Train, the meta-model is updated by:

$$\theta' = \theta - \beta \frac{1}{m} \sum_{j=1}^{m} \nabla_\theta \mathcal{L}_i^{val}(\hat{w}(\theta)) \tag{2.4}$$

where β is the learning rate for the meta-model. Actual-Train is the last step where the base DNN model is optimized taking into account the previously updated meta-model.

$$w' = w - \alpha \frac{1}{n} \sum_{i=1}^{n} \mathcal{V}_i^{tra}(\theta') \nabla_w \mathcal{L}_i^{tra}(w) \tag{2.5}$$

w' becomes the w in Eq. (2.3) for the $(t+1)$-th iteration.

2.4.2 Meta Feature Re-Weighting (MFRW)

Human attention is the ability of the brain to selectively concentrate on one aspect of the environment while ignoring other information. Attention for a DNN is a mechanism that tries to mimic the cognitive attention of the human brain, calculating a soft (or hard) mask which is then multiplied with the visual features of the network. The mask W is usually the output of a function g of some input x

$$W = g(x) \tag{2.6}$$

and W is element-wise multiplied with a feature f of the network

$$f_{att} = W \odot f \tag{2.7}$$

where $\odot$ is the symbol for element-wise multiplication.

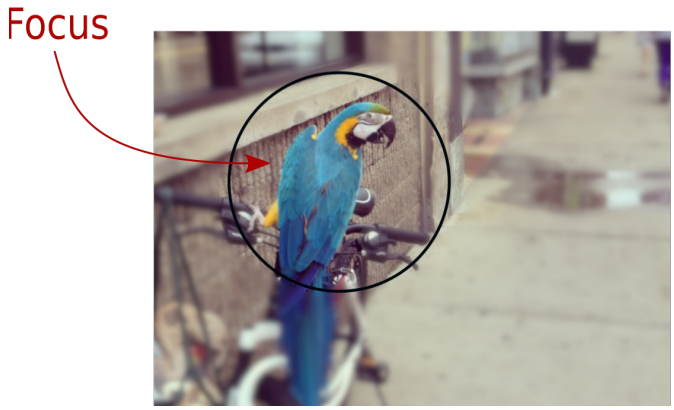

Figure 2.3: Example of how the human brain selectively concentrates on the meaningful aspect of the environment.

This intensifies the important parts of the feature and reduces the rest. We proposed a meta-attention mechanism, called Meta Feature Re-Weighting (MFRW), that can be used to mitigate noisy labels problems in the training data. A mismatch between the content of the image and its annotation can lead to the degradation of the classifier's performance for that annotated class. With our method, the main network can use only parts of the erroneous visual information to enhance performance in that class. Finding the handwritten function g that generates the right masks is very challenging. The type of noise is not always known in advance and may vary for each class. We used a meta-model to automatically infer the correct g. Meta-learning allowed the learned g to change during the training of the main network and adapt automatically to the noise present in the annotations. The element-wise product is done between the feature f extracted from a DNN and a vector of weights W_f learned from a meta-model.

$$f_{att} = W_f \odot f \tag{2.8}$$

The meta-model can take into consideration important aspects for each training data, so it can generate the proper activation weights based on them. Attention must be differentiated between the various categories, as each may have different noise levels. This is done by giving as input to the meta-model visual features extracted from the classifier's backbone. In addition, misla-

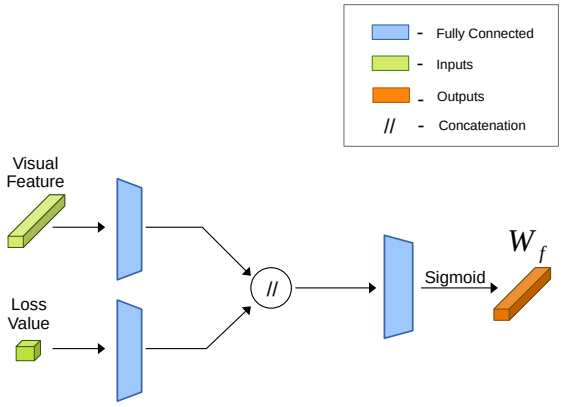

Figure 2.4: The architecture of the meta-model Ψ. The color green is used to highlight the inputs of the model. Instead, the orange indicates the attention mask output vector.

beled images have a different difficulty than cleanly labeled images, as they are outside the correct distribution of each category. The attention should be adjusted according to the difficulty that a training example represents for the classifier. This way, it can focus differently on the information in the data by learning a better representation. The cost value typically used to express the difficulty of classification samples [33] is given to the meta-model in combination with the visual features.

2.4.3 Meta model architecture

Our meta-model Ψ has a really simple architecture, as shown in Figure 2.4. The inputs of the network are a visual feature vector f and a loss value $\mathcal{L}_x$. Each input is projected in a fixed size embedding space through a separate fully connected layer. Then the embeddings are concatenated and passed to another fully connected layer that projects them into a larger common space. Its size is the sum of the dimension of each previous embedding. Finally, a linear layer elaborates the data from the common space to a vector with a size equal to the visual feature f given as input. Because the output must be an attention weight in the range $\in (0, 1)$, we added a sigmoid activation after the last layer.

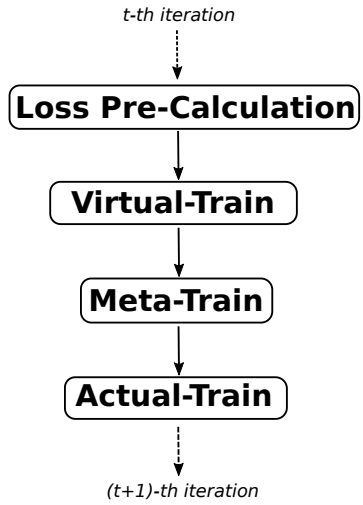

Figure 2.5: Scheme of all the steps of the optimization phase of MFRW, reaching the $(t+1)$-th iteration from the (t)-th.

2.4.4 Algorithm

In this section, we describe how the base classifier Φ and our meta-model Ψ are trained jointly. Because the meta-model needs as input the visual feature, we separate the main model $\Phi(\cdot; w)$ in two different parts: the backbone $\Phi_b(\cdot; w_b)$ and the category predictor $\Phi_c(\cdot; w_c)$. The first one has an image x as input and gives out a feature vector f. Instead, the second part has f as input and a probability score vector z as output. In this way, it is possible to manipulate the feature f directly with our meta-model Ψ. The meta-model takes two different input $\Psi(f, \mathcal{L})$ and gives back the weight vector W_f. Our algorithm is divided into 4 main phases shown in Figure 2.5. We describe each phase in detail starting with the t-th iteration and moving forward each step until we reach the $(t+1)$-th.

Different from the meta-learning optimization strategy described in 2.4.1, we need an additional initial phase, called Loss Pre-Calculation (Figure 2.6). The value of loss $\mathcal{L}^{pre}$ related to the training batch X^{train} must be calculated at the beginning. This loss value must be dependent on the original feature f^{train} and not on the weighted one f^{att}. Since this is a direct loss inference without gradient calculation, it is not an expensive step.

In the second step Virtual-Train (Figure 2.7), Φ_b^t and Φ_c^t are the virtual

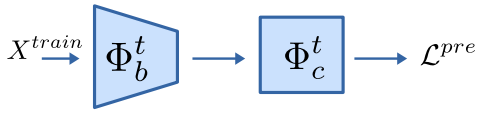

Figure 2.6: Full scheme of the Loss Pre-Calculation step. Here is computed value of loss $\mathcal{L}^{pre}$ related to the training batch X^{train}.

clones of the backbone $\Phi_b(\cdot; w_b)$ and the category predictor $\Phi_c(\cdot; w_c)$ at the beginning of the t-th iteration. We obtain the features f^{train} passing through Φ_b^t the batch X^{train}. Then the loss values $\mathcal{L}^{pre}$ pre-calculated and its relative feature f^{train} are given to Ψ^t (the meta-model at time t) in order to obtain the two vector of weights W_f and s_k. The feature f^{train} is multiplied element-wise with W_f to get a new feature vector with attention f^{att} as in section 2.4.2. The modified feature is passed to the predictor Φ_c^t obtaining the score z^{train}. Then Φ_b^t and Φ_c^t parameters are virtually updated to minimize $\mathcal{L}^{train}$, excluding those of Ψ^t.

For the third step Meta-Train (Figure 2.8), we need a clean and balanced meta dataset that will be used to train the meta-model Ψ. We pass a meta batch X^{meta} through the virtually updated Φ_b^{t+1} and Φ_c^{t+1} in order to get a validation loss $\mathcal{L}^{meta}$. In this step, the feature is not modified. Then only Ψ^t is updated minimizing $\mathcal{L}^{meta}$. In this way, the meta-model is optimized to help the main model minimize its error on clean and balanced data. Here the optimization takes into consideration also the previous Virtual-Train.

In the last phase, Actual-Train (Figure 2.9) the original Φ_b^t and Φ_c^t are optimized taking into account the updated meta-model Ψ^{t+1}. Our meta-model is used only during the training of the main network Φ when external help is needed to solve the noisy label problem. It will be discarded at test time when only the main network is retained as the final model.

2.5 Experiments

To demonstrate the effectiveness of our method, we conducted experiments on synthetically generated datasets with controlled noise structure and level.

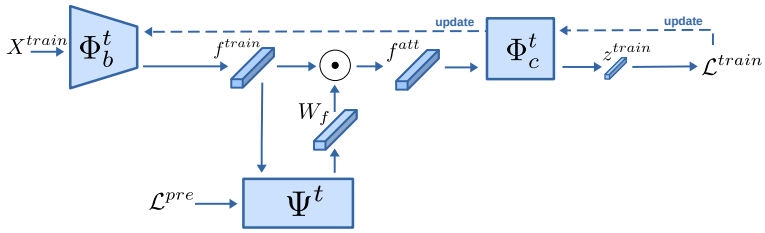

Figure 2.7: Full scheme of Virtual-Train step. The virtual clones Φ_b^t and Φ_c^t are updated minimizing $\mathcal{L}^{train}$.

Then we tested its ability to generalize with experiments on a real-world dataset.

2.5.1 Datasets

Flip CIFAR10 and Flip CIFAR100

Following previous works [26, 52, 57], we used CIFAR-10 and CIFAR-100 which are the classical choice to generate synthetic datasets containing different types of noise structures. They are composed of 50K training images and 10K test images of size 32×32. Of the training set, 1000 images with clean labels, 10 for each category, are selected to form the meta-set for meta-training. In literature Flip (or asymmetric) is a noise designed to mimic the structure of labels replaced by similar classes, e.g. dog↔cat. We decided to test our method on that type of noise because it usually appends that the label error could depend on the ambiguity between classes and similar visual patterns [70]. We created a synthetic version of CIFAR-10 and CIFAR-100. The noise ratio was controlled by a parameter p, which represents the probability that a clean example is contaminated by noise. In this way, we could test our method on different levels of noise, from $p = 0.0$ (no noise) to $p = 0.8$ (heavy noise). We also introduced Flip2 and Flip3, two new noise versions of Flip. The difference from Flip is that the noise is equally distributed over multiple similar classes, respectively two and three.

Clothing1M

In addition to synthetic datasets, we tested our method on a collection of data containing real-world label noise. Clothing1M [70] is a dataset that is

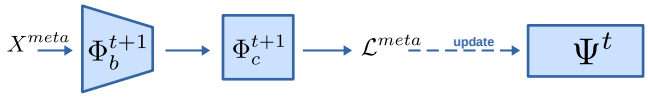

Figure 2.8: Full scheme of Meta-Train step. The meta-model is updated taking into account the previously optimized virtual clones Φ_b^{t+1} and Φ_c^{t+1}.

composed of 1 million images of clothing taken from online shopping websites. There are 14 categories like T-shirts, Shirts, Knitwear, etc. The annotations are obtained from the description of clothes images provided by online sellers, thus from non-expert annotators. This procedure originates an unpredictable noise into the image labels (Figure 2.1). The validation set of $7k$ clean data was used as the meta-set.

2.5.2 Meta-model implementation details

In every experiment, the meta-model was optimized with Adam [30] and a learning rate of 1e-4. The size of each embedding space was set always to 100.

2.5.3 Flip label noise results

To train our model under Flip (or asymmetric) label corruption noise, we used the same experimental settings of other works that studied this type of noise. The backbone was a Resnet-32 trained through SGD with a momentum of 0.9, weight decay of 5e-4, batch size of 128, and a starting learning rate of 0.1. Learning rate decreases to its $\frac{1}{10}$ at the 50 epoch and 70 epoch. The training procedure was stopped after 100 epochs.

Table 2.1 shows the accuracy results obtained on the test set of CIFAR-10 and CIFAR-100 datasets. The compared methods were directly reported with the results obtained in their paper. For MW-Net [57] and the direct training (CrossEntropy), we also reported our implementation results, denoting them with the † symbol. The accuracy gains over the other methods were significant especially on heavy levels of noise($p = 0.6$ and $p = 0.8$). From Table 2.1, our MFRW method outperforms MW-Net [57] and the ba-

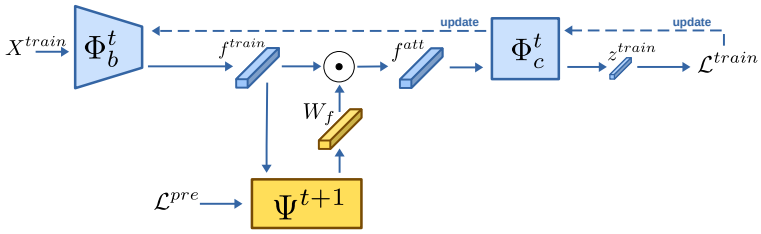

Figure 2.9: Full scheme of Actual-Train step. The original Φ_b^t and Φ_c^t are optimized with help of the updated meta-model Ψ^{t+1}.

Table 2.1: Top-1 accuracy on CIFAR10 and CIFAR100 dataset with Flip noise. The backbone used is a ResNet-32. p denotes the different levels of noise. The results for the cited method are reported directly from their original papers. Instead, † indicates the results obtained by our implementation. The first and the second best results are respectively marked with bold and underline.

Dataset	Flip CIFAR-10					Flip CIFAR-100				
Noise p	0.0	0.2	0.4	0.6	0.8	0.0	0.2	0.4	0.6	0.8
CrossEntropy [57]	92.89	76.83	70.77	-	-	70.50	50.86	43.01	-	-
Reed-Hard [50]	92.31	88.28	81.06	-	-	69.02	60.27	50.40	-	-
S-Model [17]	83.61	79.25	75.73	-	-	51.46	45.45	43.8	-	-
Self-paced [33]	88.52	87.03	81.63	-	-	67.55	63.63	53.51	-	-
Focal Loss [38]	<u>93.03</u>	86.45	80.45	-	-	70.02	61.87	54.13	-	-
Co-teaching [18]	89.87	82.83	75.41	-	-	63.31	54.13	44.85	-	-
D2L [42]	92.02	87.66	83.89	-	-	68.11	63.48	51.83	-	-
Fine-tuning [57]	**93.23**	82.47	74.07	-	-	**70.72**	56.98	46.37	-	-
MentorNet [26]	92.13	86.3	81.76	-	-	70.24	61.97	52.66	-	-
L2RW [52]	89.25	87.86	85.66	-	-	64.11	57.47	50.98	-	-
GLC [21]	91.02	89.68	<u>88.92</u>	-	-	65.42	63.07	**62.22**	-	-
MW-Net [57]	92.04	90.33	87.54	-	-	70.11	64.22	58.64	-	-
CrossEntropy†	92.33	90.56	86.25	26.67	13.58	70.18	**65.02**	50.25	18.67	4.32
MW-net† [57]	92.19	<u>90.74</u>	87.63	<u>42.41</u>	<u>27.19</u>	<u>70.57</u>	64.13	51.23	<u>19.89</u>	<u>7.42</u>
MFRW	91.87	**91.09**	**90.26**	**89.34**	**82.47**	68.93	63.54	<u>59.07</u>	**56.13**	**20.29**

sic CrossEntropy approaches by a large margin, indicating the effectiveness of our method on the synthetic Flip noise. When there is no noise ($p = 0.0$), MFRW got worse accuracy values than a normal training with the classic softmax cross-entropy loss on both CIFAR10 and CIFAR100. It happens because the advisor network, trying to help the classifier, introduces a bias

of the examples distribution contained in the meta-set. If the training distribution already reflects the test one better than the one contained in the meta-set then, introducing this meta-bias, the accuracy will be a little worse than without.

Table 2.2: Accuracy result on CIFAR10 and CIFAR100 dataset with Flip2 noise. p denotes the different level of noise. † indicates the results obtained by our implementation. The first and the second best results are respectively marked with bold and underline

Dataset	Flip2 CIFAR-10				Flip2 CIFAR-100			
Noise p	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
CrossEntropy†	<u>90.71</u>	87.83	75.83	11.86	<u>64.91</u>	57.7	36.55	7
MW-net† [57]	**90.93**	<u>88.83</u>	<u>86.85</u>	<u>27.49</u>	**65.37**	**59**	<u>36.97</u>	<u>7.99</u>
MFRW	90.66	**89.72**	**87.75**	**73.83**	63.07	<u>57.96</u>	**45.35**	**22.41**

Table 2.3: Result for Flip3 noise on CIFAR10 and CIFAR100 dataset. p denotes the different level of noise. † indicates the results obtained by our implementation. The first and the second best results are respectively marked with bold and underline

Dataset	Flip3 CIFAR-10				Flip3 CIFAR-100			
Noise p	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
CrossEntropy†	90.13	88.44	82.31	20.34	<u>65.29</u>	<u>59.35</u>	44	<u>11.07</u>
MW-net† [57]	**90.56**	<u>88.49</u>	<u>85.65</u>	<u>22.69</u>	**65.33**	**62.74**	<u>45.77</u>	10.33
MFRW	<u>90.31</u>	**88.96**	**87.73**	**75.53**	62.98	59.08	**52.28**	**25.72**

Table 2.2, 2.3 show respectively the result for noise of type Flip2 and Flip3. We can see how our method performs better than the others, especially in very noisy situations.

2.5.4 Real-world label noise results

With the real-world noise Clothing1M dataset, we used as backbone a ResNet-50 pre-trained on ImageNet. It was trained through SGD with a momentum of 0.9, weight decay of 1e-3, and a starting learning rate of 0.01. The batch size was 32, and it was preprocessed through resizing the image to 256×256, cropping the center 224×224, and performing normalization. The learning rate is divided by $\frac{1}{10}$ at 10 and 15 epochs. The complete learning process was executed for a total of 20 epochs.

Table 2.4: Comparison with state-of-the-art methods in test accuracy (%) on Clothing1M dataset with real-world noise. Results for cited methods were copied from original papers.

Method	Accuracy (%)
CrossEntropy [57]	68.94
F-correction [47]	69.84
JoCoR [69]	70.30
S-adaptation [17]	70.36
M-correction [2]	71.00
MLC [68]	71.06
Joint-Optim [61]	72.16
MLNT [36]	73.47
P-correction [73]	73.49
MW-Net [57]	73.72
MentorMix [25]	74.30
FaMUS [71]	74.43
DivideMix [35]	74.76
AugDesc [46]	75.11
MFRW	**75.35**

Table 2.4 shows the results on the Clothing1M dataset. Our method outperformed the current state-of-the-art result on real-world noise.

2.5.5 Qualitative Meta Feature Re-Weighting (MFRW) results

This section provides a qualitative analysis of various aspects of our MFRW method. To better understand how our method is helping the main classifier, it is important to look at what and how the meta-model learns.

First, we checked how the weights, learned by the meta-model, were distributed. After the last epoch of the training on Flip ($p = 0.6$) label noised CIFAR10, we extracted the first two main components of a PCA reduction on the predicted weights W_f. The figure 2.10 shows two separate large clusters which indicate that the meta-model learned to weigh the examples that contain label errors differently from those that have correct labeling. This is the effect of giving the advisor network the loss value of each training data.

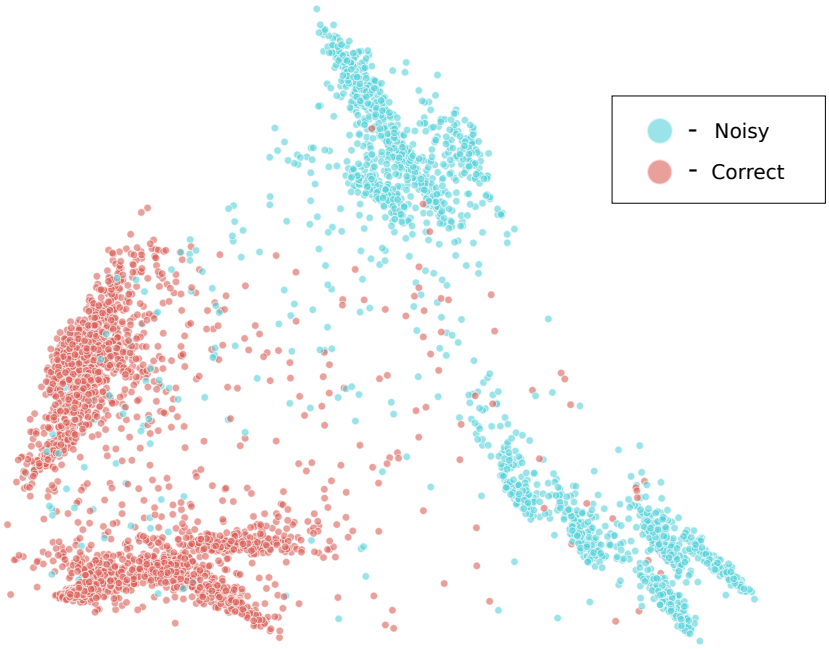

Figure 2.10: Plot of the first two main components of a PCA reduction on the weight W_f obtained from the training on CIFAR10 with Flip ($p = 0.6$) noise. Pink dots indicate an example with the correct label, instead, the light blue ones are for example with the noisy label. The clear separation between noisy and correct examples indicates a different way of weighing between these two categories

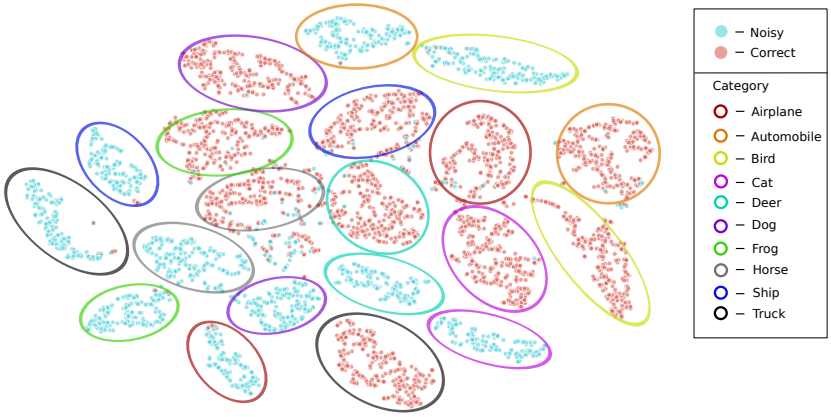

Figure 2.11: T-SNE of the first two main components of the PCA shown in Figure 2.10. Pink dots indicate an example with the correct label, instead, the light blue ones are for example with the noisy label. Each category is denoted by a different color and marked in the plot with an ellipse. Besides a separation between noisy/correct examples, there is also one at the category level. This indicates distinct predicted weight vectors W_f for features belonging to different classes.

An out-of-distribution example has a bigger cost than a good one.

Next, we did a T-SNE [63] on the two previously extracted PCA components to see if there is also a per-class separation on the weights W_f. From the T-SNE plot in figure 2.11, it is possible to deduce that the weights have also an additional per-class separation concerning the noisy/correct one. This is due to the contribution of the input visual features to the meta-model, which allows predicting different weights not only based on the loss value, but also thanks to the image content. In Figure 2.12, the weights W_f relative to the first 50 examples of the class "airplane" are shown. The information from noisy label examples (light blue border) is manipulated differently from one of the correct examples (pink border).

Figure 2.12: Attention weights W_f relative to the first 50 examples of the class "airplane" learned by our meta-model at the end of training on Flip ($p = 0.6$) noised CIFAR10. The pink color indicates examples with the correct label, instead, the light blue is for the noisy ones.

2.6 Conclusions

In this chapter, we introduced our new Meta Feature Re-Weighting (MFRW) method, which mitigated the problem of training DNNs on corrupted labels exploiting the novel concept of advisor network. We empirically show the effectiveness of our method on a synthetic and real-world noisy dataset for the classification task. The experimental results demonstrate that advisor strategy can leverage information present in noisy data helping the main network to achieve a better generalization performance. Our approach yields state-of-the-art performance on the real-world noisy Clothing1M dataset. Future research in this area may include adapting the advisor network to different problems than noise, like long-tail learning.

Chapter 3

Meta-learning advisor networks for long-tail and noisy labels image classification

Deep neural networks (DNNs) for image classification are prone to performance reduction and overfitting when trained on datasets plagued by noisy or imbalanced labels. Weight loss methods tend to ignore the influence of noisy or frequent category examples during the training, resulting in a reduction of final accuracy and, in presence of extreme noise, even a failure of the learning process. A new advisor network is introduced to address both imbalance and noise problems, able to pilot learning of a main network by adjusting the visual features and the gradient with a meta-learning strategy. In a curriculum learning fashion, impact of redundant data is reduced while recognizable noisy label images are downplayed or redirected. The proposed method is first tested on synthetic versions of CIFAR10 and CIFAR100, and then on the more realistic Imagenet-LT, Places-LT, and the Clothing1M datasets, reporting state-of-the-art results. [1]

[1]The part of this chapter has been published as "Meta-learning advisor networks for long-tail and noisy labels in social image classification" in *ACM Transactions on Multimedia Computing, Communications, and Applications, 2022* [55].

3.1 Introduction

Training deep neural networks on large amounts of labeled images is critical for modern image classification [32]. Labeling such a large amount of images has a very high cost and covering many concepts is very challenging. Therefore, automatic methods for label generation have recently been investigated by researchers, for instance in the form of semi-supervised. They exploit labeled images from non-experts (e.g., from social networks) or even unlabeled ones that are available in very large quantities at no cost. Due to their nature, these data have unbalanced or mislabeled samples [1], as shown in Figure 3.1. The large number of parameters from which deep neural networks are composed provides great adaptability in learning different tasks and concepts. This property leads to the generation of highly discriminative models if the training data are balanced and correct. When this assumption is not true and the data are unbalanced or their annotations are noisy, then there is a resulting reduction in performance and possible overfitting [10, 25]. Recent methods have attempted to address the label noise problem by measuring network confidence during training through curriculum learning [33], employing another co-trained network [18], or directly estimating noise in the set [21]. Finding samples out of the correct distribution and trying to reduce their impact on training is the general idea for dealing with this dataset problem. Instead, the unbalanced distribution (long-tail) of concepts is solved through feature augmentation techniques [8], but above all with the design of ad-hoc loss functions for this type of problem [10, 51, 60]. In contrast, a meta-learning approach is here proposed to address long-tailed and noisy labels problems, which is based on an advisor network trained to help the main classifier model (Figure 2.2) to perform better at the image classification task. During the training of a standard classification model, the advisor network adjusts feature activations and gradients of the main model by observing its feature activations and training loss. At test time, the advisor is discarded keeping only the main network as the final model. Compared to the teacher-learner paradigm our advisor network is trained to help another model instead of being trained to do image classification.

3.2 Related Works

Imbalanced training labels

Training on imbalanced (or Long-tailed) datasets is an active research field in computer vision [5, 8, 24, 51, 60, 78, 80]. A common solution present in literature is re-sampling. While [6, 19, 43] sampled more (over-sampling) training data from the minority classes to balance the distribution of all classes, [13] removed (under-sampling) data from frequent classes to make the data distribution more uniform. Under-sampling is infeasible in extreme long-tailed datasets, where the imbalance ratio between the head and the tail class is high, because most of the examples would be excluded from training. Another solution is re-weighting, where weight was assigned to each different training sample, according to its importance. [23, 67] used the inverse of class frequency to determine the weight value. Re-weighting can be done even on a sample level. In [38] a modulating loss factor was introduced to make the neural network cost-sensitive, reducing the loss contribution from easy examples. Instead, [5, 10, 29, 78] manipulated the loss based on the category distribution. In [51] an unbiased softmax function was derived to explicitly model the class distribution shift and minimize the generalization error bound. [60] introduced a new loss that avoids discouraging gradients for the rare class. [8, 74] exploited the feature augmentation method to transfer the feature variance of common classes to the rare ones. The solution proposed by [40] adopted a memory module to augment the rare categories with semantic feature representation obtained from common ones. Instead, our method exploits a new layer of meta-attention to direct the classifier's attention much more to the rare categories, while still not forgetting the common ones. Moreover, our advisor network automatically modifies the classifier's gradients to avoid the negative impact of the common classes on the rare ones.

Noisy training labels

In literature, numerous works deal with the problem of noisy labels in training data. It has been shown that the performance of machine learning systems degrades in the presence of label noise [45], [48]. One category of solutions involves a loss correction to mitigate the effect of noisy samples on the classifier network. For example, GLC [21], Reed [50], M-correction [2], F-correction [17] and S-adaptation [47] estimated the matrix of corruption probabilities to change the wrong annotations to the correct ones. Instead, [61], [42], [73] modeled the noise distribution linearly combining the output of the network and the noisy label to estimate true labels. Another

Figure 3.1: Examples of mislabeled (red cross) and correct (green tick) clothing images obtained from online shopping stores. Annotations are obtained from the description of the clothes provided by the sellers.

different approach is assigning a weight to each sample. Lower weights avoid the contribution of the correspondent sample to the training of the network. In this way, it is possible to assign low values to noisy examples and high values to correct ones. MentorNet [26] and MentorMix [25] estimated the latent weights with data-driven curriculum learning and student-teacher paradigm. Other contributions include data augmentation strategies like Mixup [76], Advaug [7] and DevideMix [35]. In contrast, our method takes advantage of an advisor network that alters activations and gradients of the main classifier can increase its generalization performance, without isolating noisy label samples from the clean ones.

Meta learning

Meta-learning was used to assist the training and optimization of learning models, and it was also applied to the long-tailed classification task. In [24,36] the meta-model learned to assign higher weights to the examples of the rare classes. With small clean validation data, the meta-model learns how to correct the biased training dataset. Meta-learning was also applied to address

the problem of the noisy labels [36, 52, 68]. For example, L2R [52] weighed each sample giving less importance to the noisy one. MLNT [36] simulates regular training with synthetic noisy labels, instead MLC [68] estimates the noise transition matrix. MW-Net [57] automatically determined an explicit weighting function that can be easily fitting to a different type of task, and it works on both the noise and imbalance training issues. These methods took advantage of a small clean validation dataset to apply the meta-learning scheme. Differently from them, our advisor network modifies the network activations using a meta-attention layer and simultaneously learns to weigh training gradients to increase the performance of the main classifier during its training. Our method addresses both imbalance and noise training concurrently.

3.3 Contributions

Our contributions are:

- Following the principles of our advisor network, we present a new meta-model to solve concurrently both the imbalance and label noise problems for the image classification task.
- We apply the Meta Feature Re-Weighting (MFRW) method, which automatically generates an attention mask on the visual features of the classifier, to the long-tail learning problem.
- The Meta Equalization Softmax (MES) activation function has been formulated to automatically adjust the grader gradients so that its learning is not adversely affected by an image belonging to a frequent category or with a noisy label.
- The effective performance of our method is shown numerically and qualitatively by experiments conducted on synthetic (long-tailed and noisy label corruption) and real-world datasets. We achieve the state-of-the-art result on Clothing1M.

3.4 Proposed Method

We developed a new advisor network that helps the deep neural network (DNN) to address both the noisy labels and long-tail image classification

problems. Our method is composed of two main parts that can work jointly: Meta Feature Re-Weighting (MFRW) and Meta Equalization Softmax (MES). The first component, MFRW, makes use of an auxiliary advisor network that automatically learns how to weigh the features extracted from a DNN during its training. Our idea was to exploit an attention mechanism to enhance the useful parts of visual information and lower the rest. If a network can concentrate only on some convenient parts of an image, that information can contribute to increasing the overall generalization capacity of the network even if the annotation is wrong. This is also true for the long-tailed distribution of data where information from common classes can be leveraged to improve performance on the rare ones. In the second component, MES, the advisor network automatically learns how to reduce the discouraging gradients of some images concerning others. In long-tailed distributions, the discouraging gradients of frequent categories samples can worsen the learning of the rare ones [60]. This can happen even with noisy labels because the discouraging gradients of an example with the wrong label can affect the correct learning of the entire classifier. These two methods can be used simultaneously to help the learning of an image classifier, reducing the negative effect that unbalanced or noisy annotations in the training datasets can produce. Our advisor network is trained with the meta-learning paradigm, so it can know the current state of the classifier and learn how to help it at that moment.

We proceed to show in detail each part of our method: Meta Feature Re-Weighting is specified in Section 3.4.1 and Meta Equalization Softmax in Section 3.4.2. Finally, the learning process of the classifier together with the advisor network is described in Section 3.4.4.

3.4.1 Meta Feature Re-Weighting (MFRW)

Deep Neural Networks can exploit the human brain mechanism that selectively concentrates the attention on one aspect of the information while ignoring others. Mimic the cognitive attentioncan be done calculating a soft (or hard) mask which is then multiplied with the visual features of the network. The mask W is usually the output of a function g of some input x

$$W = g(x) \tag{3.1}$$

and W is element-wise multiplied with a feature f of the network

$$f_{att} = W \odot f \tag{3.2}$$

where $\odot$ is the symbol for element-wise multiplication.

We proposed a meta-attention method, called Meta Feature Re-Weighting (MFRW), that mitigates noisy or imbalanced labels problems in the training data. A possible mismatch between the content of the image and its associated label can lead to a classifier's performance degradation for that annotated class. Our method made the main network use only parts of the erroneous visual information to improve performance in that class. Instead in the case of imbalance, MFRW can attribute to the visual information relative to common classes smaller importance than those of the rare ones. Finding the handwritten function g that generates the right masks for each of the two cases is challenging. We used a meta-model to automatically infer the correct g. This gives two properties to g: it can change during the training of the main network and it can adapt automatically to the problem present in the training data. The element-wise product is done between the feature f extracted from a DNN and a vector of weights W_f learned from a meta-model

$$f_{att} = W_f \odot f \tag{3.3}$$

The meta-model can take into consideration important aspects for each training data, so it can generate the proper activation weights based on them. Attention must be differentiated between the various categories, as each may have a different number of examples and noise levels. This is done by giving as input to the meta-model visual features extracted from the classifier's backbone. In the case of unbalanced labels, examples of the more common categories, since they are presented many more times during training, are easier to be learned than those of the rarer ones. In addition, mislabeled images have a different difficulty than cleanly labeled images, as they are outside the correct distribution of each category, and their size is often smaller than the correct ones. The attention should be adjusted according to the difficulty that a training example represents for the classifier. This way, it can focus differently on the information in the data by learning a better representation. The cost value typically used to express the difficulty of classification samples [33] is given to the meta-model in combination with the visual features.

3.4.2 Meta Equalization Softmax (MES)

The conventional loss function for image classification is the softmax cross-entropy. A multinomial distribution p over C categories is obtained from the network outputs score z with the softmax activation function. Then the cross-entropy is calculated between p and target distribution y. The softmax cross-entropy loss can be formulated as:

$$\mathcal{L}_{SCE} = -\sum_{j=1}^{C} y_j log(p_j) \tag{3.4}$$

where the distribution p_j is described as follow:

$$p_j = softmax(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{C} e^{z_k}} \tag{3.5}$$

When the distribution of categories in the training dataset is imbalanced, the softmax cross-entropy loss makes the learning of rare categories easily suppressed by the common ones. In [60] a softmax equalization loss is proposed to avoid the discouraging gradients from samples of frequent categories for the rare ones. The difference with the softmax cross-entropy is the weighting of a term within the softmax activation function. The new distribution p_j is calculated as:

$$p_j = softmax_{EQ}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{C} \tilde{w}_k e^{z_k}} \tag{3.6}$$

where

$$\tilde{w}_k = 1 - \beta T_\lambda(f_k)(1 - y_k) \tag{3.7}$$

The element $T_\lambda(f_k)$ is an handcrafted threshold function which outputs a value $\in \{0, 1\}$ based on the category frequency value f_k. When T_λ output is 1 the gradient is ignored, otherwise it is taken into account. Instead, β is a Bernoulli random variable with a probability of ρ to be 1 and $1 - \rho$ to be 0.

The strategy of avoiding the discouraging gradients can be useful in other problems different from imbalance training, for example image classification with noisy labels. The discouraging gradients of a mislabeled image can be scaled to not harm the correct learning of the classifier model. This behavior can be obtained by modifying the weights $\tilde{w}_k$ passed to the $softmax_{EQ}$

function in the Eq. 3.6. The element that determines each category weight is $T_\lambda(f_k)$, but it works only for the imbalance training problem. Writing a new function for the noisy annotation problem is hard because the noise can be really complex or completely unknown, for example when data is collected automatically [70].

Inspired by this we proposed a meta-learned equalization loss (MES) that can adapt the weight to the task that needs to be solved. The new formulation of the weights $\tilde{w}_k$ passed to the $softmax_{EQ}$ is:

$$\hat{w}_k = 1 - \beta s_k(1 - y_k) \tag{3.8}$$

where s_k is the vector of value $\in (0, 1)$. This vector s_k is the output of a meta-model trained to help the main model handle noise and imbalance label problems present in the training data. The visual feature and the cost of each training data are given as input to the meta-model. This allows the model to generate output vectors s_k that are differentiated between classes and between "easy" and "hard" examples.

3.4.3 Meta-model architecture

Since both MFRW and MES require the same input data, it was possible to combine the two methods through a single meta-model. Our meta-model Ψ is a neural network composed only by a fully connected layer. Its architecture is really simple. The inputs are a feature f and a loss value $\mathcal{L}_x$. Each input is projected in a fixed size embedding space through a separate fully connected layer. These embeddings are concatenated to form a larger common space, the size of which is the sum of the dimension of each previous embedding. MFRW method requires a weight vector W_f in the range $\in (0, 1)$ of the same size of the feature input f. Instead, MES needs a weight vector in the range $\in (0, 1)$ but with a length equal to the number of classes C. From the last embedding space, we get the needed outputs thanks to a fully connected layer followed by a sigmoid activation function for each of the outputs. In this way, MFRW and MES share the same internal representation learned from the inputs. The meta-model architecture permits the two methods to work together, allowing both of their benefits to be leveraged at the same time.

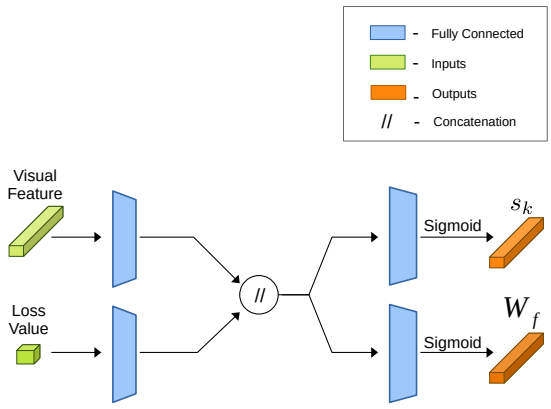

Figure 3.2: The architecture of the meta-model Ψ. The color green is used to highlight the inputs of the model. Instead, the orange indicates the attention mask output vector. This version of the meta-model has two outputs respect to the one showed in Figure 2.4.

3.4.4 Algorithm

In this section, we illustrate the four phases optimization strategy to jointly train the base classifier Φ and our new meta-model Ψ.

Firstly, we separated the main model $\Phi(\cdot; w)$ into two different parts to extract the visual features needed as input by the meta-model. The backbone $\Phi_b(\cdot; w_b)$, that takes an image x and gives out the visual feature vector f, and the classifier $\Phi_c(\cdot; w_c)$, that has f as input and a probability score vector s as output. The meta-model takes two different input $\Psi(f, \mathcal{L})$ and gives back two vectors of weights W_f and s_k. We focus on the beginning of the t-th optimization iteration and describe each step to reach the $(t+1)$-th. Unlike the meta-learning optimization strategy illustrated in 2.4.1, we added an initial phase, called Loss Pre-Calculation (Figure 2.6), to estimate in advance the value of loss $\mathcal{L}^{pre}$ related to the training batch X^{train}. This loss value relies on the original feature f^{train}.

The second step is called Virtual-Train (Figure 3.3). In this phase, Φ_b^t and Φ_c^t are respectively virtual clones of $\Phi_b(\cdot; w_b)$ and $\Phi_c(\cdot; w_c)$, taken at the beginning of the t-th iteration. The pre-calculated loss values $\mathcal{L}^{pre}$ and its

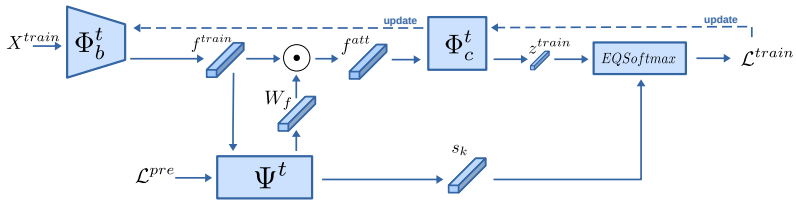

Figure 3.3: Full scheme of Virtual-Train step with both MFRW and MES methods. The virtual clones Φ_b^t and Φ_c^t are updated minimizing $\mathcal{L}^{train}$.

relative feature f^{train}, extracted from Φ_b^t, are given to the meta-model Ψ^t to get the two vector of weights W_f and s_k. As described in section 3.4.1, the feature f^{train} is multiplied element-wise with W_f to get feature vector with attention f^{att}. The altered feature vector f^{att} is given to Φ_c^t in order to obtain the score z^{train}. Then passing the vector s_k to the Eq. 3.8 of the equalization loss, described in section 3.4.2, we compute the $\mathcal{L}^{train}$. The parameters of Φ_b^t and Φ_c^t are virtually updated to minimize $\mathcal{L}^{train}$. In this phase, Ψ^t is not optimized.

The third step is called Meta-Train (Figure 2.8) because it is the meta-model Ψ is updated. We pass a meta batch X^{meta}, sampled from a clean and balanced meta-set, through the virtually updated Φ_b^{t+1} and Φ_c^{t+1} to obtain a validation loss $\mathcal{L}^{meta}$. The visual features, used in this step, are not altered by the meta-model, and the loss value is obtained with a classic softmax cross-entropy. We only update Ψ^t minimizing $\mathcal{L}^{meta}$, leaving the rest unchanged. The meta-model is optimized to help the main model minimize its error on clean and balanced data, considerating also the previous Virtual-Train.

In the last phase, called Actual-Train, the original Φ_b^t and Φ_c^t are optimized taking into account the updated meta-model Ψ^{t+1}. Our meta-model will be discarded at test time, and it is used only during the training of the main network Φ when external help is needed to solve noisy or imbalance label problems.

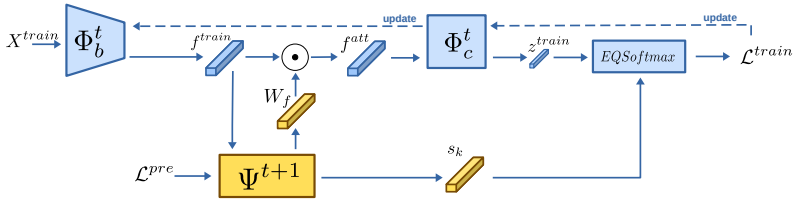

Figure 3.4: Full scheme of Actual-Train step with both MFRW and MES methods. The original Φ_b^t and Φ_c^t are optimized with help of the updated meta-model Ψ^{t+1}.

3.5 Experiments

To demonstrate the effectiveness of our method, we conducted several experiments on synthetically generated datasets with a controlled level of noise and imbalance. We also tested it in real-world datasets to prove its ability to adapt to any context.

3.5.1 Datasets

CIFAR10 and CIFAR100

Following previous works [26, 52, 57], we used CIFAR-10 and CIFAR-100 as bases to generate synthetic datasets. They are composed of 50,000 training images and 10,000 test images of size 32×32. Off the training set, we randomly selected 10 images per class to create the meta set for meta-training. The long-tailed versions of the datasets, CIFAR-LT-10 and CIFAR-LT-100, are created randomly removing training examples. Following the standard evaluation protocol for the long-tailed problem, we studied five different imbalance factors (IFs) of 200, 100, 50, 20, and 10, where IF=1 coincides with the original datasets. These IFs are related to the parameter $\mu \in (0, 1)$, where the number of examples dropped from the y-th class is $n_y \mu^y$ and n_y is the original number of training examples for that class. We tested our method even on a noisy labels version of CIFAR-10 and CIFAR-100. We chose the standard Flip (or asymmetric) noise because it is designed to mimic the structure where labels are only replaced by similar classes, e.g., dog↔cat. This type of noise usually happens when there is ambiguity between cate-

gories or visual similarity between images [70]. The noise ratio is controlled with a parameter p, which represents the probability that a correct label is flipped with the correspondent similar one. In this way we could test our method on different levels of noise, from $p = 0.0$ (no noise), to $p = 0.6$ (heavy noise). Merging the two strategies to inject data issues, we also introduced a new synthetic version of each dataset as a new evaluation protocol for the case of training data with simultaneously unbalanced and noisy labels.

ImageNet-LT

In [40] a long-tailed version of ImageNet-2012 [11] called ImageNet-LT, was introduced as standard evaluation protocol for the long-tailed problem. From a Pareto distribution with shape value $\alpha = 6$, each class size is sampled to obtain the corresponding number of images for each one. ImageNet-LT has $115,800$ training images in 1000 classes with an imbalance factor of $1280/5$. We randomly selected 10 images per class from the provided validation set to create our meta set for meta-training. The test set is the original balanced ImageNet-2012 validation set with 50 images per class.

Places-LT

The Places-LT dataset is created by sampling from the dataset Places-2 [77] with the same strategy used for ImageNet-LT. The training set is composed of 62.500 images from 365 classes with an imbalance factor of $4980/5$. The test set has 100 images per class. Our meta set is created randomly selected 10 example per class from a validation set of 20 images per class.

Clothing1M

The Clothing1M [70] is a dataset that is composed of 1 million images of clothing taken from online shopping websites. There are 14 categories like T-shirts, Shirts, Knitwear, etc. The labels are obtained from the text of the images provided by the sellers and not from an expert annotator. This process introduces into the labels a real-world noise, which cannot be predicted in advance. In Figure 3.1 some examples of noisy and correct annotations are shown. A validation set of $72,409$ manually well-annotated images is provided and it was used as the meta dataset in our experiments.

3.5.2 Meta-model implementation details

In every experiment, the meta-model was optimized with Adam [30] and a learning rate of 1e-4. The size of each embedding space was set always to 100. The probability of ρ of the Bernoulli distribution β for MES was equal to 0.9.

3.5.3 Long-Tail label distribution results

We conducted several experiments on the imbalance training problem related to image classification. We tried different settings and datasets to compare our method with the others in the literature. We tested MFRW and MES both disjointly and together (MFRW-MES).

CIFAR-LT-10 and CIFAR-LT-100

The first part of experiments on CIFAR-LT-10 and CIFAR-LT-100 was conducted with a Resnet-32 network trained through SGD with a momentum of 0.9, weight decay of 5e-4, batch size of 128, and a starting learning rate of 0.1. The learning rate decreased to its 1/10 at the 160 epoch and 180 epoch, stopping the learning at 200 epochs. In Table 3.1 the results in this experiment's setting of different works are shown.

Instead, in the second part of the experiments, a more strong preprocessing and a different learning rate scheduler were applied to the training. The results of Table 3.2 were obtained with a Resnet-32 trained for 13000 iteration with a batch of 512, on which was applied AutoAugment [9] and Cutout [12]. The initial learning rate was 0.1, then decreased to zero with a Cosine Annealing scheduler [41]. The optimizer used was SGD with a momentum of 0.9, weight decay of 5e-4. Here only MFRW-MES was compared against the other methods.

We also tested our method on a model with fewer parameters (ResNet-18) but with the same settings as for the experiments done in Table 3.1. The results are shown in Table 3.3.

From Table 3.1, 3.2 and 3.3 is possible to notice how our method exceeds or is in line with the results obtained from the state-of-the-art algorithms for long-tailed training. We obtained the best accuracy values in almost all IFs, especially when the dataset is extremely unbalanced (IF=200,100,50). Table 3.3 shows the effectiveness of our method even on smaller network architecture. Both MFRW and MES obtained good results, even individually. We

Table 3.1: Test accuracy (%) of ResNet-32 architecture on CIFAR-LT-10 and CIFAR-LT-100 under different imbalance factors (IFs). The results for the cited methods are reported directly from their original papers. Instead, † indicates the results obtained by our implementation. The first results are marked in bold and the second ones with an underline.

Dataset	Long-Tailed CIFAR-10					Long-Tailed CIFAR-100				
IF	200	100	50	20	10	200	100	50	20	10
CrossEntropy [57]	65.68	70.36	74.81	82.23	86.39	34.84	38.32	43.85	51.14	55.71
Focal Loss [38]	65.29	70.38	76.71	82.76	86.66	35.62	38.41	44.32	51.95	55.78
Fine-tuning [57]	66.08	71.33	77.42	83.37	86.42	38.22	41.83	46.4	52.11	57.44
CB Loss [10]	68.89	74.57	79.27	84.36	87.49	36.23	39.6	45.32	52.59	57.99
L2RW [52]	66.51	74.16	78.93	82.12	85.19	33.38	40.23	44.44	51.64	53.73
MW-Net [57]	68.91	75.21	80.06	84.94	87.84	37.91	42.09	46.74	54.37	58.46
LDAM-DRW [5]	-	77.03	-	-	88.16	-	42.04	-	-	58.71
CE [24]	-	76.41	80.51	86.46	88.85	-	43.35	48.53	55.62	59.58
LDAM [24]	-	80.00	82.34	84.37	87.40	-	44.08	49.16	52.38	58.00
FaMUS CE [71]	-	79.30	83.15	87.15	89.39	-	45.60	49.56	56.22	60.42
FaMUS LDAM [71]	-	80.96	83.32	86.24	87.90	-	46.03	49.93	55.95	59.03
BALMS† [51]	74.76	80.42	83.56	87.33	**89.19**	42.91	**47.21**	51.85	**57.43**	**61.61**
MFRW	**78.07**	80.43	**84.08**	**87.43**	88.76	40.77	44.85	49.65	56.46	60.25
MES	72.23	78.35	81.84	86.71	88.95	40.56	44.68	50.81	57.07	61.35
MFRW-MES	75.91	**81.19**	83.87	86.84	88.83	**43.33**	46.8	**52.02**	56.95	60.6

could observe how they could be used simultaneously without compromising the final performance of the classifier.

ImageNet-LT and Places-LT

Following the experiment setup of [51], we employed ResNet-10 and ResNet-152 networks for ImageNet-LT and Places-LT, respectively. For ImageNet-LT, we adopted an initial learning rate of 0.2 and decayed with Cosine Annealing scheduler during training of 180 epochs. For Places-LT, the learning rate started at 0.005 and it was reduced like for ImageNet-LT. We trained ResNet-152 for a total of 60 epochs with a batch size of 64. In both cases, our method started from a baseline that had been pre-trained on the entire dataset. We did not freeze the feature extractor part of the pre-trained network as the decoupled training strategy of [27] does. We pre-trained the backbone to accelerate the total training time and to make our method starts from an almost good feature extractor.

Table 3.4 shows the result of MFRW-MES on ImageNet-LT and Places-LT. In the first dataset, our method achieved a Top-1 accuracy value comparable to other methods that only target this task. Instead, for the Places-LT dataset, we got the second-best result.

With these experiments, we showed how our algorithm can solve the

Table 3.2: Test accuracy (%) of ResNet-32 architecture on CIFAR-LT-10 and CIFAR-LT-100 under different imbalance factors (IFs). Autoaugment and Cutout are additionally applied as preprocessing on the data. The results of the cited methods are reported directly from their original papers. Bold is used for the first results and underline for the second ones.

Dataset	Long-Tailed CIFAR-10			Long-Tailed CIFAR-100		
IF	200	100	10	200	100	10
CrossEntropy	71.2	77.4	90.0	41.0	45.3	61.9
CBW	72.5	78.6	90.1	36.7	42.3	61.4
CBS	68.3	77.8	90.2	37.8	42.6	61.2
Focal Loss [38]	71.8	77.1	90.3	40.2	43.8	60.0
CB Loss [10]	72.6	78.2	89.9	39.9	44.6	59.8
LDAM Loss [5]	73.6	78.9	90.3	41.3	46.1	62.1
Equalization Loss [60]	74.6	78.5	90.2	43.3	47.4	60.5
cRT [27]	76.6	82.0	91.0	44.5	50.0	63.3
LWS [27]	78.1	83.7	<u>91.1</u>	45.3	<u>50.5</u>	63.4
BALMS [51]	**81.5**	<u>84.9</u>	**91.3**	<u>45.5</u>	**50.8**	63.0
MFRW	78.32	82.49	90.22	42.9	47.05	62.77
MES	77.12	81.19	91.03	43.52	48.44	<u>63.63</u>
MFRW-MES	<u>81.38</u>	**84.97**	90.99	**46.52**	50.44	**64.06**

Table 3.3: Classification accuracy (%) of a low parameters architecture, ResNet-18, trained on the same settings of Tab.3.1. The first and the second best results are respectively marked with bold and underline. † indicates the results obtained by our implementation.

Dataset	Long-Tailed CIFAR-10					Long-Tailed CIFAR-100				
IF	200	100	50	20	10	200	100	50	20	10
CrossEntropy	70.22	75.16	82.32	87.24	90.73	38.87	43.65	48.55	57.09	62.59
CB Loss [10]	69.16	75.16	81.9	86.61	90.79	38.58	43.51	48.15	57.02	63.1
FSA [8]	77.06	80.57	84.51	88.54	**91.75**	42.84	46.57	51.9	58.69	65.08
BALMS† [51]	76.86	<u>81.78</u>	85.28	**89.27**	90.86	42.19	<u>48.07</u>	<u>53.83</u>	59.87	64.13
MFRW	<u>79.49</u>	81.35	<u>86.32</u>	<u>89.23</u>	<u>91.44</u>	<u>43.19</u>	47.51	53.15	<u>60.38</u>	**65.36**
MES	73.11	77.96	83.33	88.69	90.63	40.28	44.49	50.1	58.24	63.99
MFRW-MES	**79.94**	**83.43**	**86.8**	89.13	91.02	**43.85**	**50.04**	**54.12**	**61.37**	<u>65.28</u>

long-tail data problem via a simple advisor network trained with the meta-learning paradigm.

Table 3.4: Top-1, Top-3 and Top-5 accuracy (%) of ResNet-10 classifier on Imagenet-LT and Places-LT. We report directly the result of the cited methods from their original papers. The first and the second results are marked with bold and the second ones with underline.

Dataset	Imagenet-LT			Places-LT		
Method	Top-1	Top-3	Top-5	Top-1	Top-3	Top-5
CrossEntropy	25.26	38.65	47.88	27	47.95	58.56
RCB [24]	29.9	<u>46.71</u>	54.82	30.8	<u>52.05</u>	<u>62</u>
OLTR [40]	35.6	-	-	35.9	-	-
Equalization Loss [60]	36.44	-	<u>61.19</u>	-	-	-
cRT [27]	<u>41.8</u>	-	-	36.7	-	-
LWS [27]	41.4	-	-	37.6	-	-
BALMS [51]	**41.8**	-	-	**38.7**	-	-
MFRW-MES	41.78	**59.87**	**67.25**	<u>38.34</u>	**61.2**	**71.09**

3.5.4 Flip label noise results

We trained our model under Flip (or asymmetric) label corruption noise at various levels. To assess the performance of the advisor network, we compared it to other works that studied this type of noise. We trained a Resnet-32 through SGD with a starting learning rate of 0.1 and batch size of 128. We decreased the learning rate at epoch 50 and 70 by a factor of 0.1. We stopped the training after 100 epochs. We also reproduced the [51] algorithm under this experiment setting to observe how an ad-hoc long-tailed distribution method works under the Flip noise.

We can notice from Table 3.5 that our method obtained the best results for the flip noise on CIFAR10 and CIFAR100. The use of our advisor network avoided a drastic accuracy drop than the other methods, especially when the noise was really strong ($p = 0.6$). When there is no noise ($p = 0.0$) our method got worse accuracy values than a normal training with the classic softmax cross-entropy loss on both CIFAR10 and CIFAR100. It happens because the advisor network, trying to help the classifier, introduces a bias of the examples distribution contained in the meta set. If the training distribution already reflects the test one better than the one contained in the meta set then, introducing this meta bias, the accuracy is a little worse than without.

Table 3.5: Test accuracy on CIFAR10 and CIFAR100 dataset with Flip (asymmetric) label noise. The backbone used is a ResNet-32. p denotes the different levels of noise. The results for the cited methods are reported directly from their original papers. Instead, † indicates the results obtained by our implementation. The first and the second best results are respectively marked with bold and underline.

Dataset	Flip CIFAR-10				Flip CIFAR-100			
Noise p	0.0	0.2	0.4	0.6	0.0	0.2	0.4	0.6
CrossEntropy [57]	92.89	76.83	70.77	-	70.50	50.86	43.01	-
Reed-Hard [50]	92.31	88.28	81.06	-	69.02	60.27	50.40	-
S-Model [17]	83.61	79.25	75.73	-	51.46	45.45	43.8	-
Self-paced [33]	88.52	87.03	81.63	-	67.55	63.63	53.51	-
Focal Loss [38]	<u>93.03</u>	86.45	80.45	-	70.02	61.87	54.13	-
Co-teaching [18]	89.87	82.83	75.41	-	63.31	54.13	44.85	-
D2L [42]	92.02	87.66	83.89	-	68.11	63.48	51.83	-
Fine-tuning [57]	**93.23**	82.47	74.07	-	**70.72**	56.98	46.37	-
MentorNet [26]	92.13	86.3	81.76	-	70.24	61.97	52.66	-
L2RW [52]	89.25	87.86	85.66	-	64.11	57.47	50.98	-
GLC [21]	91.02	89.68	88.92	-	65.42	63.07	62.22	-
MW-Net [57]	92.04	90.33	87.54	-	70.11	64.22	58.64	-
CrossEntropy†	92.33	90.56	86.25	26.67	70.18	65.02	50.25	18.67
MW-Net† [57]	92.19	90.74	87.63	42.41	<u>70.57</u>	64.13	51.23	19.89
BALMS† [51]	92.86	90.99	83.51	51.76	69.66	65.61	56.83	39.16
MFRW	91.87	91.09	90.26	89.34	68.93	63.54	59.07	56.13
MES	<u>93.03</u>	<u>91.25</u>	**90.76**	**90.58**	69.74	**65.36**	**62.96**	**60.82**
MFRW-MES	92.46	**91.44**	<u>90.7</u>	<u>90.21</u>	68.33	<u>65.17</u>	<u>62.26</u>	<u>58.43</u>

3.5.5 Long-Tail & Flip label noise results

We decided to introduce a new synthetic dataset setting in which unbalanced and noisy label problems are both present. We chose 3 values of IFs (200, 100, 10) and two of p (0.4, 0.6), and all possible combinations for both CIFAR10 and CIFAR100 were generated. All experiments were performed training a Resnet-32 with the same settings and hyperparameters of the one used to obtain the results listed in Table 3.1. This experiment is important to establish the ability of an algorithm to handle different types of dataset conditions at the same time.

We compared our method with BALMS [51], because it is designed for long-tailed distributions, and with MW-Net [57], which can deal with any type of bias in the data, similar to us. The results shown in Table 3.6 indicate that our advisor network can manage at the same time both noisy labels and

Table 3.6: Test accuracy on CIFAR10 and CIFAR100 dataset with two levels of Flip label noise p (0.4, 0.6) and three different imbalance factors IFs (200, 100, 10). The backbone used is a ResNet-32. † indicates the results obtained by our implementation of different methods. Bold is used for the first results and underline for the second ones.

Dataset	LT Flip CIFAR-10						LT Flip CIFAR-100					
Noise p	0.4			0.6			0.4			0.6		
I.R	200	100	10	200	100	10	200	100	10	200	100	10
CrossEntropy†	49.64	56.98	76.58	31.78	31.96	31.78	22.03	23.81	39.48	12.08	13.65	19.6
MW-Net† [57]	45.74	52.43	82.22	32.06	33.22	46.5	24.34	25.24	39.05	12.96	14.96	20.01
BALMS† [51]	53.73	59.24	70.4	48.53	52.55	57.39	27.02	29.18	44.44	18.8	22.14	32.02
MFRW	**55.9**	63.52	83.76	44.41	52.62	69.16	23.45	25.26	38.08	17.65	18.48	29.58
MES	53.33	**64.51**	**85.8**	44.45	52.46	83.17	25.41	26.76	47.54	17.12	18.79	38.75
MFRW-MES	55.13	61.67	85.34	**50.19**	**59.38**	**90.26**	**31.73**	**34.2**	**53.89**	**21.79**	**24.3**	**39.99**

long-tailed distributions better than the other methods.

3.5.6 Real-world label noise results

In order to test real-world noise, we used Clothing1M and ResNet-50 as backbone, pre-trained on ImageNet, that was trained through SGD with a momentum of 0.9, weight decay of 1e-3, and a starting learning rate of 0.01. The batch had a size of 32 and it was preprocessed resizing the image to 256×256, then random cropping a 224×224 patch, and finally performing normalization. The total training process consisted of 20 epochs where the learning rate was multiplied by 0.1 after 10 and 15 epochs.

The results reported in Table 3.7 show how our method obtains the state-of-the-art accuracy on the clothing dataset, improving it by $3,10\%$ compared to the best algorithm previously used [46].

3.5.7 Qualitative long-tail advisor network results

Since section 2.5.5 discussed the MFRW method in the case of noisy labels, we analyzed the W_f learned on CIFAR-LT-100 when the IF is 200, the most difficult setting case of the long-tail problem. The predicted weights W_f differ both between different classes. The weights of the frequent class "apples" (Figure 3.5) have more values closer to zero (black color) instead of the one belonging to the rare class "roses" (Figure 3.6) with a lot of value close to one (white color). This means that the information from common category examples is ignored much more than the one belonging to the rare classes.

Table 3.7: Comparison with state-of-the-art methods in test accuracy (%) on Clothing1M dataset with real-world noise. Results for cited methods were copied from original papers.

Method	Accuracy (%)
CrossEntropy [57]	68.94
F-correction [47]	69.84
JoCoR [69]	70.30
S-adaptation [17]	70.36
M-correction [2]	71.00
MLC [68]	71.06
Joint-Optim [61]	72.16
MLNT [36]	73.47
P-correction [73]	73.49
MW-Net [57]	73.72
MentorMix [25]	74.30
FaMUS [71]	74.43
DivideMix [35]	74.76
AugDesc [46]	75.11
MFRW	75.35
MES	<u>76.43</u>
MFRW-MES	**77.44**

Moreover, every example belonging to the same class is not weighted equally. This is shown in the right-top section of Figure 3.6 where there are some weight vectors with more value close to one (white color) than others. It happens because those examples contain information that is still useful to the main classifier.

Softmax weights learned with MES

We investigated how our softmax weight s_k, learned with MES, differs from the handmade solution proposed by [60]. We measured the effectiveness of each solution by calculating the Mean Absolute Error (MAE) between the distribution of the classes size, normalized between 0 and 1, and the vector of the weights passed to Eq 3.6. Figure 3.7 shows the MAE values obtained during the training of the main network on CIFAR-LT-100 when the IF is 200. MES fits the target distribution better than the simple threshold function applied in [60] and does not need any extra hyperparameter tuning.

Figure 3.5: Attention weights W_f, relative to examples of the common class "apples", obtained after a complete training on CIFAR-LT-100 with IF equals to 200.

Figure 3.6: Attention weights W_f, relative to examples of the rare class "roses", obtained after a complete training on CIFAR-LT-100 with IF equals to 200.

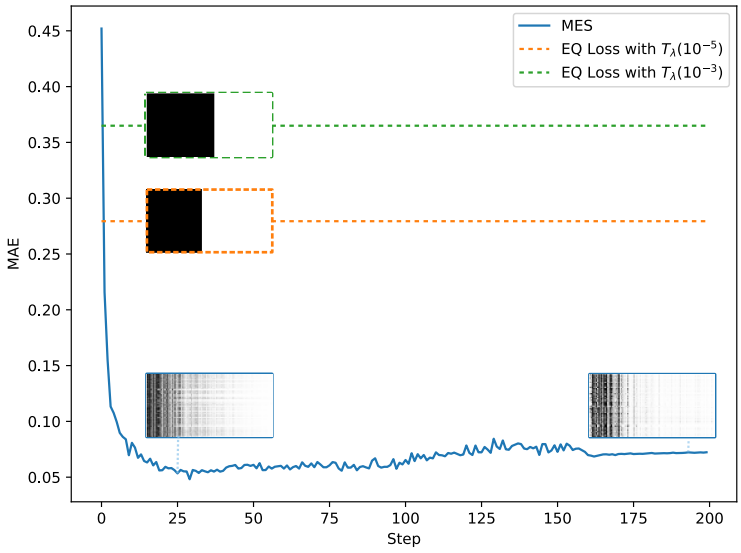

Figure 3.7: Comparison of MES with the two functions defined in [60] on CIFAR-LT-100 with a IF of 200. In the graph is reported the Mean Absolute Error (MAE) between the distribution of the size of the classes (normalized between 0 and 1) and the vector of weights given to Eq 3.6. Lower values of MAE indicate a better fit of the target distribution. In the graph, there are also details of the predicted vector weights s_k at various learning steps.

3.5.8 Qualitative long-tail & Flip advisor network results

In this section, we provide a qualitative analysis of our two methods, MFRW and MES, in the context of a training dataset with concurrently noisy annotations and long-tail label distribution. As discussed in section 2.5.5, we checked how the weights were distributed after the meta-model learning. We extracted the first two main components of a PCA reduction, shown in Figure 3.8, on the predicted weights W_f at the end of training on Flip LT CIFAR-10 with $IF = 100$ and $p = 0.4$. Unlike the result obtained in 2.10 where the two large clusters are separated, in this more complex setting the division is less evident. The effect of the loss value, given as input to the advisor network, is altered by the presence of few dominant classes, that include the majority of the training data. Out-of-distribution examples are not only data with a wrong label but also the ones belonging to rare categories.

Next, we did a T-SNE [63] on the two previously extracted PCA components to see if the per-class separation of Figure 2.11 on the weights W_f was always present. From the T-SNE plot in figure 3.9, it is possible to conclude that the weights still maintained additional per-class separation concerning the noisy/correct one. The split is less evident for rare categories than common ones, due to the significant disparity of the number of training data between them.

In Figure 3.10 and Figure 3.11 are respectively shown the attention weights W_f learned by the advisor network for the common class "apples" and the rare class "roses" from Flip LT CIFAR-100 (generated with $IF = 100$ and $p = 0.4$). In both figures, a clear distinction is present between the weights generated by the MFRW method for samples with noisy and correct labels.

We analyzed the softmax weights s_k learned by MES. We compared them with the handmade solutions proposed by [60]. To measure the effectiveness of each solution, we calculated the Mean Absolute Error (MAE) between the distribution of the classes size, normalized between 0 and 1, and the vector of the weights passed to Eq 3.6. Figure 3.12 shows the MAE values obtained during the training of the main network on Flip LT CIFAR-100 with IF of 100 and $p = 0.4$. MES fits the target distribution better than the simple threshold function applied in [60] without any extra hyperparameter tuning. Different from Figure 3.7, the predicted vector weights s_k have values close to zero even for some of the rare classes. This behavior was caused by

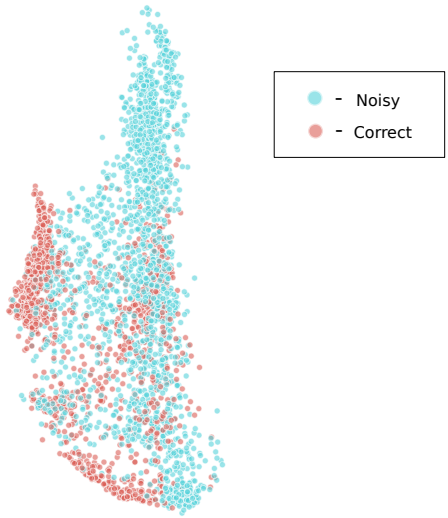

Figure 3.8: Plot of the first two main components of a PCA reduction on the weight W_f learned from the training on Flip LT CIFAR-10 with the noise parameter $p = 0.4$ and an IF of 100. Pink dots indicate an example with a correct label, instead, the light blue ones stand for example with a noisy label. The separation between noisy and correct points indicates a different way of weighing these two categories. The division is not completely clear as in 2.10 due to the presence of unbalance in the training data.

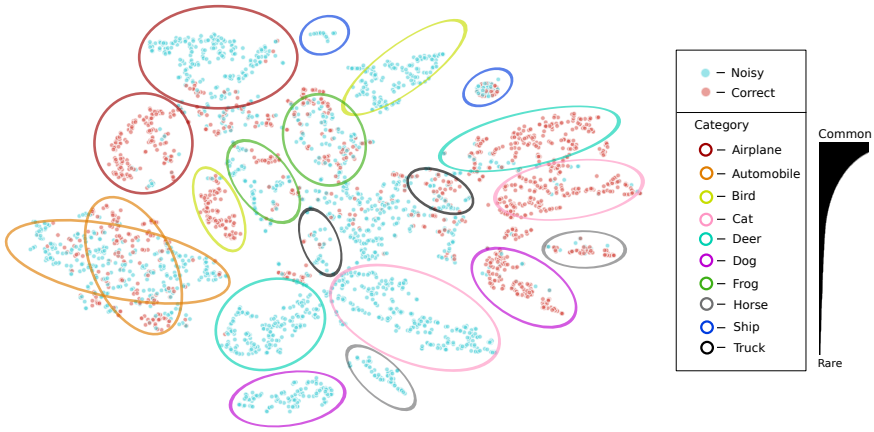

Figure 3.9: T-SNE of the first two main components of the PCA shown in Figure 3.8. Pink dots indicate an example with the correct label, instead, the light blue ones are for example with the noisy label. Each category is denoted by a different color and marked in the plot with an ellipse. The black indicator on the right indicates how the classes follow the long-tail distribution. Besides a separation between noisy/correct examples, there is also one at the category level. This indicates predicted weight vectors W_f for features belonging to different classes even if there is a unbalance between them.

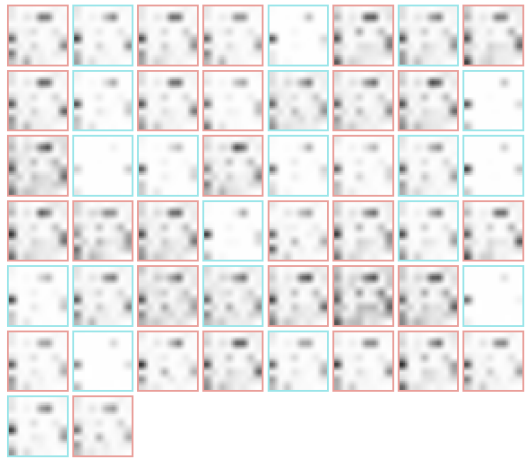

Figure 3.10: Attention weights W_f, relative to examples of the common class "apples", obtained after a complete training on Flip LT CIFAR-100 with IF equals to 100 and $p = 0.4$. The pink color indicates examples with the correct label, instead, the light blue is for the noisy ones.

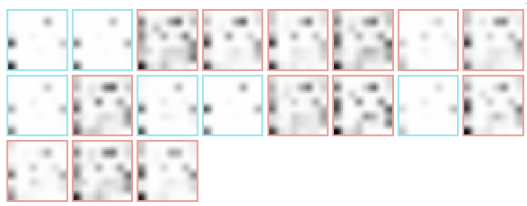

Figure 3.11: Details of the attention weights W_f, relative to examples of the rare class "roses", obtained after a complete training on Flip LT CIFAR-100 with IF equals to 100 and $p = 0.4$. The light blue color denotes the noisy data, instead the pink color is used for examples with the correct label.

the presence of Flip noise inside the image annotations. Our method MES permitted discouraging gradients on some rare categories to help the main classifier handle the noisy annotations problem.

3.6 Conclusions

We introduced two methods Meta Feature Re-Weighting (MFRW) and Meta Equalization Softmax (MES), that make use of a novel concept of advisor network to mitigate the problem of training DNNs on noisy labels and long-tailed class distributions. We empirically showed the effectiveness of our method on synthetic generated and real-world datasets for the classification task. Experimental results demonstrate that the advisor strategy can help the main classifier achieve better generalization performance for both the training data problems. We introduced a new synthetic dataset setting where the long-tailed distribution is mixed with the noisy label problem. Then we showed how our method succeeds in solving both problems simultaneously unlike other similar work. We got the state-of-the-art performance on the Clothing1M dataset, which contains real-world label noise. Future research in this area may include adapting the advisor network to a more complex task than classification, like Object Detection or Image Segmentation.

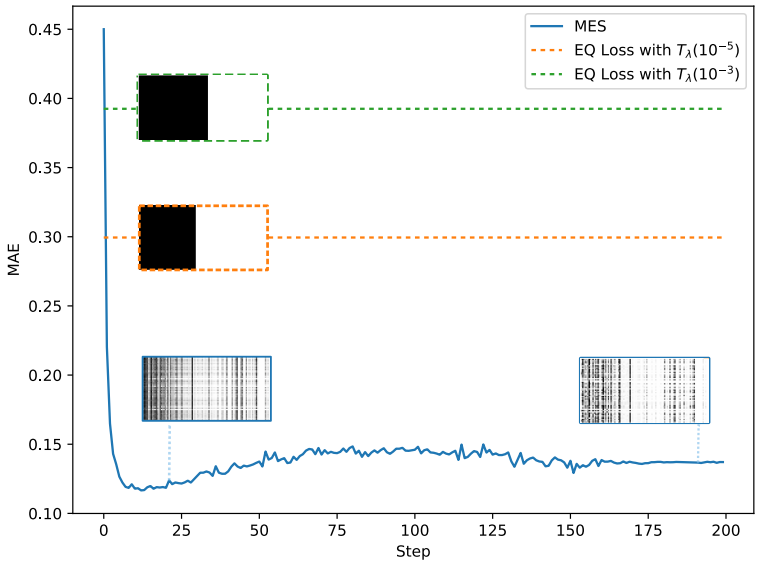

Figure 3.12: Comparison of MES with the two functions defined in [60] on LT Flip CIFAR-100 with an IF of 100 and a noise level p equals to 0.4. On the y-axis is shown the Mean Absolute Error (MAE) between the distribution of the size of the classes (normalized between 0 and 1) and the vector of weights s_k. The x-axis is the actual training epoch. Lower values of MAE indicate a more reasonable fit of the noisy long-tail target distribution. Details of the predicted vector weights s_k are exhibited at various learning steps.

Chapter 4

Conclusion and Future Challenges

In this thesis, we studied the problems of noisy label learning and long-tail learning for the image classification task. The methods we developed employ the novel concept of advisor network to help the classifier address the two issues separately and even together. Through a meta-attention module and an adaptive activation function, the advisor network adjusts the visual features and gradients generated during the training of a classifier.

To address the noisy label problem, we presented in Chapter 2 a novel solution that mimics the human attention capacity of the brain to focus only on some portions of the viewed scene. A soft weight mask was multiplied with the visual feature of the main classifier during its learning. We exploited a meta-learning approach, where a meta-model automatically generates the correct weight mask thanks to a small verified and balanced auxiliary dataset. This learned attention permitted the classifier to focus only on the useful information of samples with noisy annotations. We demonstrated its effectiveness against other related methods over synthetic and real-world datasets. We showed that our method is capable of handling high levels of labels noise, obtaining the state-of-the-art result on the Clothing1M dataset.

In Chapter 3, we employed the concept of advisor network to alleviate the problem of training DNNs for image classification on noisy labels and long-tailed class distributions. The advisor guided the learning of a classifier adjusting its visual feature and gradients with a meta-attention method and a meta-activation function. We presented a new artificial dataset setting

where noisy annotations are combined with the long-tailed distribution. Experiments on the synthetic and real-world datasets demonstrated the efficacy of our methods. We prove that our advisor network can work to help the classifier against high imbalance and noisy data. We got a new state-of-the-art result on the real-world noise Clothing1M dataset.

In the future, there are several directions to further investigate on. The advisor network proposed in Chapter 3, exploits an activation function to modify the discouraging gradients obtained during the training of a neural network. This property could be also applied to tasks different and more complex than image classification, like Object Detection and Image Segmentation. these two tasks have great application typically for vision solutions that interact with the real world, for instance medical imaging, self-driving, and satellite imaging. The real-world objects typically follow a long-tail distribution. Our advisor network could be adapted to interact with more complex neural network architectures. For example, the meta-attention method could interact with visual features extracted by the ROI pooling operation inside the Faster-RCNN [53] or directly with the last classification layer of YOLO [49].

Bibliography

[1] G. Algan and I. Ulusoy, "Image classification with deep learning in the presence of noisy labels: A survey," *Knowledge-Based Systems*, vol. 215, p. 106771, 2021.

[2] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," in *International Conference on Machine Learning*. PMLR, 2019, pp. 312–321.

[3] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio *et al.*, "A closer look at memorization in deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 233–242.

[4] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[5] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *Advances in Neural Information Processing Systems*, vol. 32, pp. 1567–1578, 2019.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[7] Y. Cheng, L. Jiang, W. Macherey, and J. Eisenstein, "Advaug: Robust adversarial augmentation for neural machine translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5961–5970.

[8] P. Chu, X. Bian, S. Liu, and H. Ling, "Feature space augmentation for long-tailed data," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, 2020, pp. 694–710.

[9] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

[10] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9268–9277.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[12] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[13] C. Drumnond, "Class imbalance and cost sensitivity: Why undersampling beats oversampling," in *ICML-KDD 2003 Workshop: Learning from Imbalanced Datasets*, vol. 3, 2003.

[14] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

[15] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational intelligence*, vol. 20, no. 1, pp. 18–36, 2004.

[16] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[17] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," 2016.

[18] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Advances in Neural Information Processing Systems*, 2018.

[19] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[20] J. Han, P. Luo, and X. Wang, "Deep self-learning from noisy labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5138–5147.

[21] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," *Advances in Neural Information Processing Systems*, vol. 31, pp. 10 456–10 465, 2018.

[22] M. E. Houle, "Local intrinsic dimensionality i: an extreme-value-theoretic foundation for similarity applications," in *International Conference on Similarity Search and Applications*. Springer, 2017, pp. 64–79.

[23] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5375–5384.

[24] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, "Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7610–7619.

[25] L. Jiang, D. Huang, M. Liu, and W. Yang, "Beyond synthetic noise: Deep learning on controlled noisy labels," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4804–4815.

[26] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2304–2313.

[27] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *International Conference on Learning Representations*, 2019.

[28] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, "Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis," *Medical Image Analysis*, vol. 65, p. 101759, 2020.

[29] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3573–3587, 2017.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] V. Koltchinskii and D. Panchenko, "Empirical margin distributions and bounding the generalization error of combined classifiers," *The Annals of Statistics*, vol. 30, no. 1, pp. 1–50, 2002.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[33] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," *Advances in neural information processing systems*, vol. 23, pp. 1189–1197, 2010.

[34] K.-H. Lee, X. He, L. Zhang, and L. Yang, "Cleannet: Transfer learning for scalable image classifier training with label noise," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5447–5456.

[35] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *International Conference on Learning Representations*, 2019.

[36] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Learning to learn from noisy labeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5051–5059.

[37] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1910–1918.

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[39] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2008.

[40] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2537–2546.

[41] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[42] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. Erfani, S. Xia, S. Wijewickrema, and J. Bailey, "Dimensionality-driven learning with noisy labels," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3355–3364.

[43] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten, "Exploring the limits of weakly supervised pretraining," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 181–196.

[44] I. Misra, C. Lawrence Zitnick, M. Mitchell, and R. Girshick, "Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2930–2939.

[45] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artificial intelligence review*, vol. 33, no. 4, pp. 275–306, 2010.

[46] K. Nishi, Y. Ding, A. Rich, and T. Hollerer, "Augmentation strategies for learning with noisy labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8022–8031.

[47] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1944–1952.

[48] M. Pechenizkiy, A. Tsymbal, S. Puuronen, and O. Pechenizkiy, "Class noise and supervised learning in medical domains: The effect of feature extraction," in *19th IEEE symposium on computer-based medical systems (CBMS'06)*. IEEE, 2006, pp. 708–713.

[49] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[50] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," *arXiv preprint arXiv:1412.6596*, 2014.

[51] J. Ren, C. Yu, S. Sheng, X. Ma, H. Zhao, S. Yi, and H. Li, "Balanced meta-softmax for long-tailed visual recognition," in *Proceedings of Neural Information Processing Systems(NeurIPS)*, Dec 2020.

[52] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4334–4343.

[53] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[54] S. Ricci, T. Uricchio, and A. Del Bimbo, "Learning advisor networks for noisy label image classification," *21st International Conference on Image Analysis and Processing (ICIAP)*, 2021.

[55] ——, "Meta-learning advisor networks for long-tail and noisy labels in social image classification," *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2022.

[56] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, "Deep learning is robust to massive label noise," *arXiv preprint arXiv:1705.10694*, 2017.

[57] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: learning an explicit mapping for sample weighting," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 1919–1930.

[58] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," *arXiv preprint arXiv:1406.2080*, 2014.

[59] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[60] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan, "Equalization loss for long-tailed object recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 662–11 671.

[61] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5552–5560.

[62] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman, "Learning from noisy labels by regularized estimation of annotator confusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 244–11 253.

[63] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[64] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 839–847.

[65] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy, "The devil of face recognition is in the noise," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 765–780.

[66] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[67] Y.-X. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 7032–7042.

[68] Z. Wang, G. Hu, and Q. Hu, "Training noise-robust deep neural networks via meta-learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4524–4533.

[69] H. Wei, L. Feng, X. Chen, and B. An, "Combating noisy labels by agreement: A joint training method with co-regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 726–13 735.

[70] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2691–2699.

[71] Y. Xu, L. Zhu, L. Jiang, and Y. Yang, "Faster meta update strategy for noise-robust deep learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 144–153.

[72] J. Yao, J. Wang, I. W. Tsang, Y. Zhang, J. Sun, C. Zhang, and R. Zhang, "Deep learning from noisy image labels with quality embedding," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1909–1922, 2018.

[73] K. Yi and J. Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7017–7025.

[74] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Feature transfer learning for face recognition with under-represented data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5704–5713.

[75] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[76] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[77] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.

[78] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9719–9728.

[79] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transactions on knowledge and data engineering*, vol. 18, no. 1, pp. 63–77, 2005.

[80] L. Zhu and Y. Yang, "Inflated episodic memory with region self-attention for long-tailed visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4344–4353.