Learning with noisy labels (LNL)

정재환 (jhjung1227@gmail.com)
TEL: 010-4828-7602

#Report 1

Model : **DivideMix** [1]

Summary

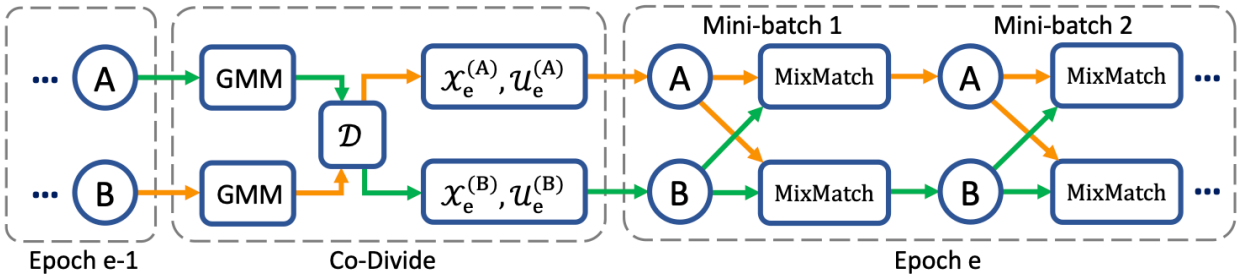
Discards the labels that are highly likely to be noisy, and **leverage** the noisy samples as unlabeled data to regularize the model in a Self-Supervised Learning(**SSL**) manner from overfitting and improve generalization performance.

Co-divide, which trains two networks simultaneously.

- For each network (ne1, ne2), by fitting a Gaussian Mixture Model (GMM) on its per-sample loss distribution , It can divide the training samples into a labeled set and unlabeled set.
- The divided data is then used to train the other network. Co-divide keeps the two networks diverged, so that they can filter different types of error and avoid confirmation bias in self-training

During SSL phase, model architecture improves MixMatch with **label Co-refinement** and **Co-guessing** to account for label noise

- For labeled samples, it refines their gound-truth labels using the network's predictions guided by the GMM for the other network.
- For unlabeled samples, it uses the ensemble of both networks to make reliable guesses for their labels.



- Train two networks (A and B) simultaneously.
- At each epoch, a network models its per-sample loss distribution with a GMM to divide the dataset into a labeled set (mostly clean) and unlabeled set (mostly noisy), which is then used as training data for the other network (i.e. co-divide)
- At each mini-batch, a network perform semi-supervised training using an improved MixMatch method.
- The reason to use **MixMatch** method is to improve **SSL methods performance**.
  SSL methods aim to improve the model's performance by leveraging unlabeled data.
  And current state-of-art SSL methods mostly involve adding an additional loss term on unlabeled data to regularize training
  (**consistency regularization** and **entropy minimization**)
  Recently, **MixMatch** and **MixUp** regularizations are recommended.

- Perform label co-refinement on the labeled samples and label co-guessing on the unlabeled samples.

Why I Choose this model

- Because this model contains many advances in this field.
  Dividing train datasets by Loss values is come from MentorNet.
  (In DivideMix, Gaussian Mixture Model is used to separate Clean and Noisy Data distribution)
  Training 2 networks simultaneously is come from Co-teaching.
  (In Dividemix, By GMM fitting, the networks can use both clean and noisy data. And By Mixup augmentation of clean and noisy data, the outputs can be used in the other network)
  Semi-Supervised Learning concepts are displaced to MixMatch Method.

  All of this things and the method of discarding labels (noisy) and get SSL caught my attention and the possibility to train the model only with the given noisy datasets are the reason.

More Detail

1. Gaussian Mixture Model

- Train two models (network A and network B) simultaneously and compute each Losses every epochs.
- These values are the quantities that are expected to be the sum of many independent processes(from ResNet34) with a nearly normal distributions.
- That's why I use Gaussian Distribution and in particular the GMM
- (I set up two distributions in GMM because I want to separate clean and noisy labeled data)
- By using Mixture of Gaussian distributions, it is possible to know how much probability each data belongs to which distribution and if the data has a higher probability in a distribution with a lower loss, it will be separated to "Clean Data" and others will be "Noisy Data". (This concepts was studied at "Memorization effect" and "MentorNet")

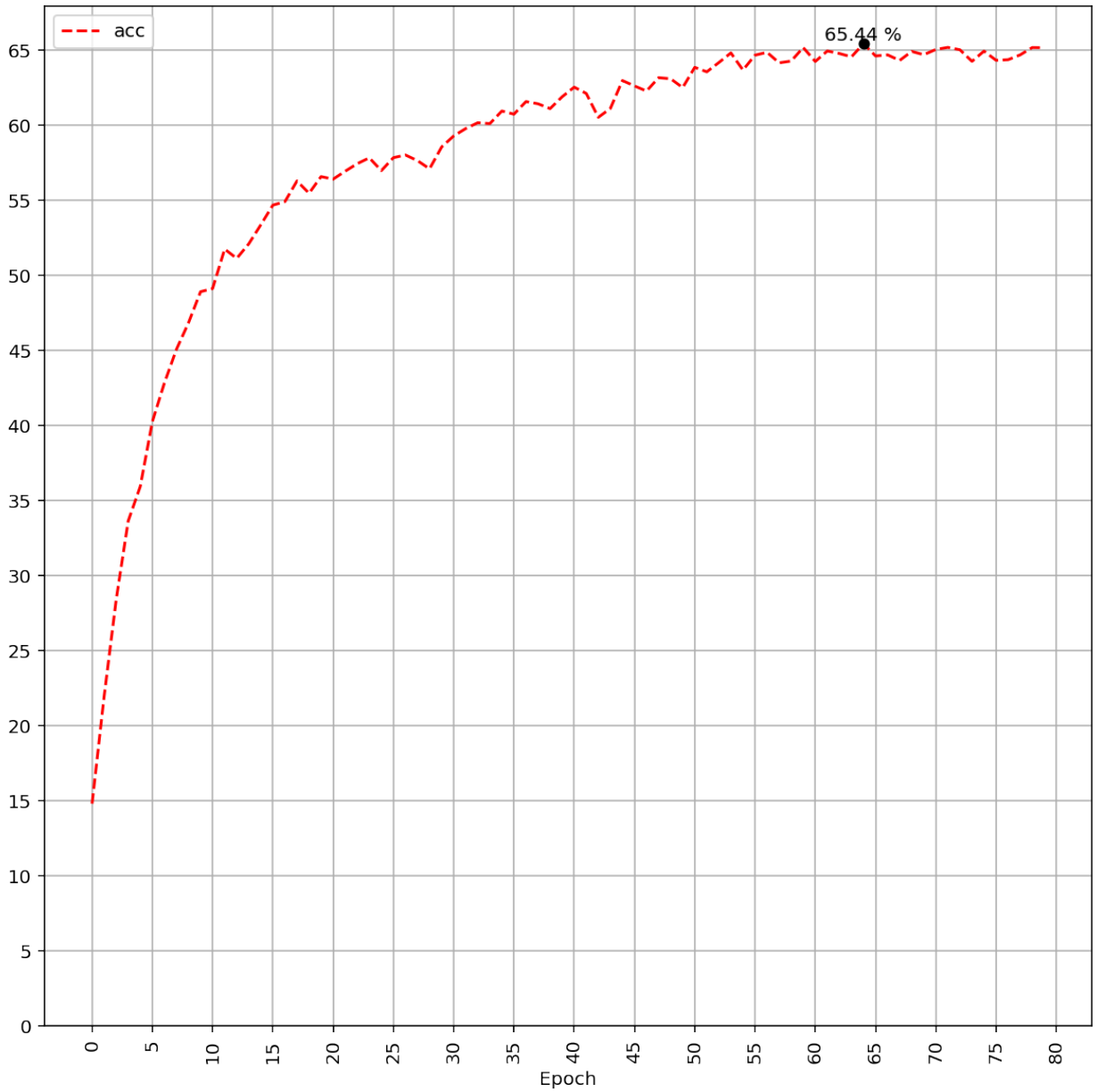
2. Co-teaching

- If only one model is used to LNL process, a serious problem arises in the process of dividing into Clean and Noisy data from the loss value.
- In MentorNet, the loss from one network will be directly transferred back to itself in the second mini-batch of data, and the error should be increasingly accumulated.
- However, in Co-teaching, since two networks have different learning abilities(they start from different initial values), they can filter different types of loss introduced by noisy labels.
- In this exchange procedure, the loss flows can be reduced by peer networks mutally.
- In summary, when we use only one model, once misclassified Clean data is not trained again on next steps, but if we use two models and do cross-update, this method is less affected by the size of the initial loss and can prevent overfitting ofspecific data.

3. MixMatch

- In DivideMix, it uses MixMatch methods during SSL phase.
- (SSL: Semi-Supervised Learning; I had an error in notation "Self")
- This MixMatch methods unifies consistency regularization, entropy minimization, and the MixUp regularization into one framework.
- ("Consistency regularization" and "Entropy minimization" are one of the methods of SSL)
    - **Consistency regularization** : Encourage the model to output same predictions on perturbed unlabeled data
    - **Entropy minimization** : Encourage the model to ouput confident predictions on unlabeled data.
    - **MixUp augmentation** : Encourage the model to have linear behaviour between samples.
- Before entering MixMatch, two tasks preceds; Co-refinement and Co-guessing.
    - Co-refinement: refining by linearly combining the label of the real data and the prediction result of the model.
    - Co-guessing: creating new label by averaging the prediction results of the two model.
- Clean data and Noisy data are augmented through MixUp augmentation, respectively.
- And after this task, these two kinds of augmented data are used to update other network.
    - Using Cross-entropy Loss about Clean data. (same as original model)
    - Using L2-Loss about Noisy data. (because these data labels were made by co-guessing)

#Report 2



I didn't do train-validation split and get training accuracy. Because I only have the train datasets which are already mixed with noisy labeled . To get training accuracy, as far as I understand, it will be possible to obtain the training data as cleaned dataset first, do divide the train-validation set, and then post-process the train dataset with noise(we will use).

When I do classification task by using only clean data (Fined Labels), I will try to separate train, validation, and test datasets. Because I know I can't use test data during training task.

#Report 3

1. Transfer Learning
   - Using Pre-trained Models trained by refined dataset(Cifar-100, ImageNet, JFT etc.).
   - Doing Fine tuning the head layers according to the number of classes.
   - Dividing samples into cleanly labeled and wrongly labeled sets by training loss.

   I think this method will be the best.
   In Divide Model, I use warm-up network which trains two networks simultaneously for 30 epochs. I think this phase can be replaced to transfer learning by bringing pre-trained models which are already trained to other similar datasets.

2. Visual Embedding
   The key point of this model(DivideMix) and many others depend on the Loss.
   They classify(or divide) the data into clean data(low loss) and noisy data(high loss).
   I think this can be replaced by 'Visual Embedding'.
   More precisely, it seems to be more effective in the initial labeling process, but I guess that it also can be helpful for LNL process.

3. Augmentation Strategies for Learning with Noisy Lablels[4]
   - Just read 'Abstract' and it caught my attention, so after this test, I will read it in detail. The method on this paper probably leads to an improvement in the performance of the model.

4. Change Student Model to others

   - I think ResNet34 is enough yet to train Cifar-100N but on this time, as many state-of-the-art models are released I just wonder if what it would be like to change this models.
     Actually I've studied about 'ViT', 'Swin-T', and 'ConvNeXt' before doing this test. So I wanted to use these models when taking this test, but gave up because of small datasets(train data: 49,999), the inability to use other datasets and transfer learning, and the possibility of overfitting(due to the given noisy labeled datasets).

5. Change Hyper parameters
   - My model started to converge from about 65 epochs.
     At first, I set the learning rate to be lowered from 150 epochs, but I think setting from 70 epochs will make better performance.

References

[1] Junnan Li, Richard Socher, and Steven C.H. Hoi. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. arXiv:2002.07394v1

[2] Lu Jian, Di Huang, Mason Liu, and Weilong Yang. Beyong Synthetic Noise: Deep Learning on Controlled Noisy Labels. arXiv preprint arXiv:1911.09781v3, 2020

[3] Lu Kiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. arXiv:1712.05055v2

[4] Kento Nishi, Yi Ding, Alex Rich, and Tobias Hollerer. Augmentation Strategies for Learning with Noisy Labels. arXiv preprint arXiv:2103.02130v3, 2021