Assignment 2

Abstract

This report aims to solve the class-based label noise problems in the current machine (deep) learning environment. Based on the previous literature work, the label noise methods with known and unknown flip rates methods and noise rate estimation methods are implemented. The conclusion and future work are stated based on the experiment and results generated by these methods mentioned.

1 Introduction

Problem identification

We are currently living in a big data era, and the demand for the large training dataset used in machine learning or deep learning is constantly increasing. However, these datasets always have the problem of class-dependent label noise. The main reasons that cause the prevailing class-dependent label noise problem are the labels are normally annotated by non-expert labelers, the labeling task could be subjective, and annotating correct labels might need more cost and time due to the lack of discriminative information.

Importance of the problems

It is important to value and solve this problem since if the model is trained on the dataset with noisy labels, it would seriously undermine the performance of the classifier, which degenerates the quality of the tasks such as image classification. We could hire skilled annotation experts to assign the label, or searching help from domain knowledge experts, but it could still take a considerable amount of time and cost. However, for larger datasets, it could be infeasible and not realistic. It shows the desperate need to have a method to help us to learn the class-dependent label efficiently.

Classifiers

In this assessment, five classifiers are built, which are Forward based CNN model, the TNT-based model, the Resnet model, and the T-matrix-based-TMTT model. The additional model is used as a baseline which is the Baseline CNN-based model. The details would be introduced later.

2 Previous works

As this assessment is dealing with the label noise in a multi-classification problem, we would focus on the multi-class problems instead of the instead of binary class problems. To solve the multi-class-dependent label noise problem, two learning methods are applied: Forward and Backward learning with the T matrix. To learn the T matrix, we have an anchor point assumption method to estimate the T matrix. And to ensure our classifiers are robust to the noise, the loss corrections method is applied.

For the forward method, the relationship between the class posterior probabilities of noisy data, the class posterior probabilities of clean data, and the t matrix (the formula of this method) is shown below [1]

$$[\mathrm{P}(\mathrm{Y}^{\sim} = 1|\mathrm{X}),\ldots,\mathrm{P}(\mathrm{Y}^{\sim} = \mathrm{C}|\mathrm{X})]^{T} = \mathrm{T}[\mathrm{P}(\mathrm{Y} = 1|\mathrm{X}),\ldots,\mathrm{P}(\mathrm{Y} = \mathrm{C}|\mathrm{X})]^{T}$$

The practice is we insert the T matrix between the SoftMax layer of the neural network and the loss function, since the final output is the noisy data class posterior probabilities, the output of the SoftMax layer is a good approximation of the clean class posterior probabilities. For the backward method, similar to the forward method, we need to times T^{-1}on both sides, the formula is shown below [1]

$$[\mathrm{P}(\mathrm{Y} = 1|\mathrm{X}),\ldots,\mathrm{P}(\mathrm{Y} = \mathrm{C}|\mathrm{X})]^{T} = T^{-1}[\mathrm{P}(\mathrm{Y}^{\sim} = 1|\mathrm{X}),\ldots,\mathrm{P}(\mathrm{Y}^{\sim} = \mathrm{C}|\mathrm{X})]^{T}$$

The forward method is more convenient to build since it does not need to calculate the inverse of the T matrix. However, we need to place restraints on the forward model after we inserted the transition matrix, which costs more computational power. The backward method is more computationally cheaper than forward learning, but it must do an extra T matrix inverse, sometimes the matrix cannot be inverted, and thus we cannot get the T matrix. Even if it is inversed, the inverse of the T matrix brings more estimation error which could decrease the accuracy of the model.

T matrix is also called flip matrix, if we have C classes, then we would have C^2 class posterior probabilities flipping between the C classes. To learn the T matrix, the anchor assumption method is applied. To get the anchor points, the instances with high i^{th} class posterior as the anchor points for i^{th} class, thus we could learn the whole T matrix using anchor point [1].

Using the anchor point assumption method helps us to learn the transition matrix more accurately, however, as we are using the high i^{th} noisy class posterior to approximate the clean i^{th} class posterior, which could cause information loss, the classifiers might degenerate. And sometimes, there are no appropriate anchor points, the transition matrix cannot be learned accurately, and the performance of the classifier would be seriously degenerated [1].

The use of the loss correctness method could build a classifier that is robust to the label noise and realize the noise learning more efficiently. However, in the practice, it cannot be fully utilized since the noise raise need to be known as priori [2].

3 Methods

Data pre-processing

The pre-processing on the three datasets is the same, we first expand the dimension of the training instances x to fit the input format that is required by the numeral network classifiers built later. For the training label y, we have applied the one-hot encoding for feeding to the models later. Finally, to make 80% of the training data used to train the model and 20% of the data for validation, the data partition is applied.

Baseline model establishment and result

CNN neural network is used to build the Baseline CNN-based model on the three datasets-FMNIST5, FashionMNIST6, and CIFAR. The model is built using a kernel size of 5, a stride size of 1, and a learning rate of 0.003. By training after 5 epochs, testing accuracy result is generated. The result of the baseline model on the three datasets is shown below

Table 1: Accuracy of the baseline model on each dataset

Dataset	Accuracy
FMNIST5	0.623
FashionMNIST6	0.633
CIFAR	0.352

The testing performance of the CNN-based baseline model on the first two datasets' clean data is represented using the accuracy and standard deviation.

As table 1 shows, we could find that the Baseline model has relatively poor performance on the first two datasets with around 0.623 accuracy value. The last dataset CIFAR has an extremely low accuracy value of 0.352. it shows that the CNN baseline model is not robust enough to handle the class-based label noise in the three datasets, and the CIFAR shows a worse performance. The reason that causes the CIFAR's worse performance is that the instances in the first two datasets are mainly hand-written images, whilst in the last datasets, they are colourful photo-like images, thus the CIFAR dataset's clean label is harder to learn. Moreover, it is also possible that the class-based-noise-label rate in the CIFAR is higher than that in the other two datasets.

3.1 Label noise methods with known flip

label noise methods' formulation

Label noise methods with known flip on the dataset FashionMINIST0.5 and 0.6 is using the forward learning method, and the T matrix is directly using the given T matrix. The formula is shown below:

$$[\mathrm{P}(\tilde{\mathrm{Y}} = 1|\mathrm{X}),\ldots,\mathrm{P}(\tilde{\mathrm{Y}} = \mathrm{C}|\mathrm{X})]^T = \mathrm{T}\,[\mathrm{P}(\mathrm{Y} = 1|\mathrm{X}),\ldots,\mathrm{P}(\mathrm{Y} = \mathrm{C}|\mathrm{X})]^T$$

For the dataset FashionMINIST0.5 and 0.6, two classifiers are built: a forward-learning based CNN model with a given T matrix classifier (Forward based CNN model) and a forward-learning based CNN model with a given T matrix and TNT loss function classifier (TNT-based model). These two models both use the forward-learning and the two given T matrixes by the two datasets to train the model and test on the clean data. The main difference is that classifier 2 applies the TNT loss function (created by us) based on classifier 1. The formula of the TNT loss function is

MAE (ESTIMATED_T- Given_T) β + BCE(Prediction-Noisy_label) (1- β)

MAE is the mean square error of the estimated t matrix and the given t matrix; BCE is the binary cross entropy of our prediction and the label with noise. TNT loss could improve based on the existing binary cross-entropy loss, so that it could assist the given matrix to build a more label noise robust classifier.

Experiments

In the experiments, the two classifiers are trained using 1000 epochs, batch size of 300, and a learning rate of 0.0001. The binary cross entropy loss function is applied to the Forward based CNN model, TNT loss, which is the combination of the binary cross entropy loss and means absolute loss has been applied to the TNT-based model. Each classifier is trained with the validation generated by random sampling 10 times. The trained models are tested on the two datasets' clean testing labels.

Results and discussion

The testing performance of Forward based CNN model and TNT-based model on the first two datasets' (FMNIST5, FashionMNIST6) clean data are represented using the accuracy and standard deviation.

Table 2: Accuracy of the Forward-based and TNT-based model on each dataset

Dataset	Forward accuracy	Forward sd	TNT accuracy	TNT sd
FMNIST5	0.9179	0.0043	0.8977	0.0044
FashionMNIST6	0.8595	0.0093	0.8081	0.0144

From table 2, we could see the Forward based CNN model has good performance on the testing data of the first two datasets, the accuracy value is from 0.86 to 0.92, with a standard deviation value of 0.004 to 0.009. It shows that our forward-based CNN model is robust to the class-based label noise (higher than the baseline model's accuracy on the first two datasets, as shown in table 1, which are 0.623 and 0.633), and the model has good stability.

We could also find that the TMT-based model has lower performance than the Forward based CNN model since the testing accuracy has decreased on the two datasets. It shows that the TMT-based model is still robust, but the application of the TNT loss has not improved and even undermined the performance of the Forward based CNN model.

3.2 Label noise rate estimation method

Our noise rate estimation method is called the Resnet-based estimator. Since the transition matrix is not given, we need to learn the transition matrix using the formula mentioned before, shown as

$$\tilde{P}^{T} * P = T * I * M.$$

It means transported noisy class posterior probabilities P̃^T times the clean class posterior probabilities P is equal to the transition matrix T times identity matrix I times the numbers of the instances for each class M.

For the calculation of the T matrix, we treat I as constant 1, the labels of our data are balanced, and we have 3 classes since we could directly divide the M with the value of 3 on the left side of the formula. Finally, the right side is only the transition matrix T, and the left side is the formula to calculate the T transition matrix.

We use Resnet to train on the noisy label data, as the Resnet neural network is proven to be robust enough regarding some research , it is robust due to its numeral network architecture as its more predictive representations, align more with the target function instead of the noise function [3]. After training the Resnet network, we assume the output is the best approximation of clean class posterior probabilities P, As the class posterior probabilities of the noisy data have been given (using the noisy label data), then we could learn the T matrix as the formula shows.

Experiments

The Resnet used to output the clean class posterior probabilities are trained using 3 epochs, a batch size of 300, a learning rate of 0.0001, and loss function binary cross entropy. Only using 3 epochs is because the Resnet is very deep and complicated, if the epochs keep increasing, the model would be easily overfitting.

Results and discussions

The transition matrix learned based on the FashionMINIST0.5 is shown below

$$\begin{matrix} 0.4697 & 0.3009 & 0.2356 \\ 0.2076 & 0.4584 & 0.3496 \\ 0.2614 & 0.1675 & 0.6177 \end{matrix}$$

The transition matrix learned based on the FashionMINIST0.6 is shown as below

$$\begin{matrix} 0.3810 & 0.3025 & 0.3197 \\ 0.3015 & 0.3719 & 0.3281 \\ 0.2619 & 0.2706 & 0.4831 \end{matrix}$$

The transition matrix learned based on the CIFAR is shown below

$$\begin{matrix} 0.3701 & 0.3435 & 0.2831 \\ 0.3149 & 0.3698 & 0.3150 \\ 0.3129 & 0.2726 & 0.4345 \end{matrix}$$

The average mean square error of each element in the matrix has been calculated for the FashionMINIST0.5 and FashionMINIST0.6, the mean squared error of the FashionMINIST0.5 is 0.009232, and which of the FashionMINIST0.5 is 0.001283. It shows that our estimated matrix is very close to the given matrix, which indicates a good estimation accuracy of our estimation method.

3.3 Label noise methods without known flip

label noise methods' formulation

There are two classifiers are built that use the label noise methods without known flip on the dataset CIFAR- Resnet without transition matrix and Resnet with transition matrix and T loss function. The first classifier Resnet model is training the Resnet neural network using the binary cross entropy loss function and using their output to approximate the clean class posterior probability (the first label noise method without known flip). Thus, the Resnet itself is a noise-robust learning method (the robustness of the Resnet is mentioned before), we do not specifically illustrate the method formula here, since it is just a normal Resnet neural network. The second classifier T-matrix-based-TMT model is applying the T matrix and TNT loss based on the Resnet model, as the second label noise learning method without known flip. The formula of this method is the same as the one we show in the t estimator part, but this time, the target is to learn and test the model which is based on the clean label data. The formula is shown below

$\tilde{P}^T * P = T * I * M.$

After training the first classifier, we could use it to estimate the T matrix (the same as how we estimate the t matrix in the previous two datasets as explained before). After obtaining the T matrix, we could use the formula above to learn the clean class posterior probabilities, the TNT loss function is applied, then we could build a noise-robust classifier.

Experiments:

In the experiments, the epochs number is only 3 for the two classifiers here, since as mentioned before, the Resnet is easily overfitting if the model keeps training for more epochs due to its deep and complex architecture. The other experiment settings are the same as the experiment of Label noise methods with known flip, which are mentioned before. The trained models are tested on the last datasets' clean testing labels.

Results and discussion

The testing performance of the Resnet model and TMT and T-matrix-based-TMT model on the last datasets' (CIFAR) clean data are represented using the accuracy and standard deviation.

From table 3, we could see the Resnet model is robust to the label noise (higher than the base-line model's accuracy on the CIFAR dataset, as shown in table 1, which is 0.352), with a slightly higher value of sd (0.05), which shows the model is not stable enough. However, after applying the t matrix and T loss function, the testing accuracy has been improved, and also the stability of the model has been improved (with a less sd value).

Table 3: Accuracy of the baseline model on each dataset

	Resnet accuracy	Resnet sd	T-matrix-TMT accuracy	T-matrix-TMT sd
CIFAR	0.676	0.0498	0.6902	0.0046

4 Conclusion

4.1 Conclusion based on the results

Based on the results discussed before, firstly, we could conclude that the application of the given T matrices (known flips) has made the classifier robust to the class-based label noise, and the Resnet model (unknown flips) is also robust to the class-based label noise by implementing the Resnet neural network architecture. Secondly, we could notice that the application of the TNT would make the model more robust to the class-based label noise when the T-matrix is given (known flips), but TNT loss could help to improve the robustness of the classifier when we use the Resnet-based estimator to estimate the T-matrix and apply it on the Resnet model with the TNT loss (unknown flips). Finally, our Resnet-based estimator has a good estimation accuracy on the first two datasets (low MSE), which means it could estimate the T matrix accurately based on a class-based label noise dataset.

4.2 Future Work

In this project, we generated two CNN-based classifiers and two Resnet-based classifiers. One thing we can improve in the future is to apply our CNN-based models to the CIFAR dataset. The reason we choose Resnet to apply on CIFAR is that our CNN-based models are not able to do so. The reason is the structure of our model is too simple and not deep enough. Besides, we can combine our TMT loss with the t estimator together to have a better estimation. Currently, our TMT loss required a T matrix as input to guide the model. However, there are some inner relationships toward the T matrix. For example, the sum of each line and column of this matrix should be equal to 1. Thus, we may be able to create a new version of TMT loss that required no T matrix as input. Instead, we can use these inner relationships to calculate the loss of the estimated T matrix.

Reference

[1] Xia, X. et al. (2019) Are anchor points really indispensable in label-noise learning?, arXiv.org. Available at: https://arxiv.org/abs/1906.00189 (Accessed: November 2, 2022).

[2] Patrini, G. et al. (2017) “Making deep neural networks robust to label noise: A loss correction approach,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Preprint]. Available at: https://doi.org/10.1109/cvpr.2017.240.

[3] Li *et al*, "How Does a Neural Network's Architecture Impact Its Robustness to Noisy Labels?" *ArXiv.Org,* 2021. Available: http://ezproxy.library.usyd.edu.au/login?url=https://www.proquest.com/working-papers/how-does-neural-networks-architecture-impact/docview/2604669201/se-2.