REPRODUCIBILITY CHALLENGE - MULTI CLASS CLASSIFICATION WITHOUT MULTI-CLASS LABELS

Team Relu: Ruihuang Ding, Bin Zhang & Yefan Zhuang
Department of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
{rd3n20,bz1u20,yz5e20}@soton.ac.uk

ABSTRACT

The purpose of this report is to explore the reproducibility of the paper proposed by Hsu et al. (2019). The original paper proposes a meta-learning classification model based on pairwise similarity. This report reproduces some of the results of that paper and presents an analysis based on the experimental results. The repository is available at: GitHub repository [1].

1 INTRODUCTION

Nowadays, Deep learning is the most powerful techniques for approximation. However, it demands a large number of labels during the training process that is difficult to collect in reality. Another problem is that the type of methods (Supervised, Unsupervised and Semi-Supervised learning) are mainly depending on the structure of the given data. Instead of solving the classification problem under various constraints, the paper (Hsu et al., 2019) proposed a solution that relaxes these limitations to a meta-problem and defines a model to study the underlying learning pattern, and validate it with different deep learning architecture and datasets under supervised, unsupervised and semi-supervised learning scenarios.

In this report, we explore the reproducibility of the result provided by the paper (Hsu et al., 2019). Note that we only pay attention to the supervised and unsupervised scenarios.

2 THE APPROACH OF THE ORIGINAL PAPER

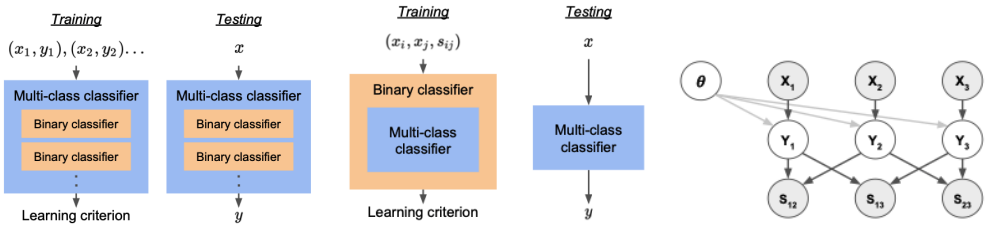
2.1 CORE IDEA

As shown in Figure 1a, the multi-classification problem is solved by taking the same task encapsulating scheme in general, where a series of ensemble binary classifiers wrapped by a multi-class classifier and with a large requirement of true labels.

To relax the limitations, the paper proposed a novel strategy that reverses the task encapsulation order. Showing in Figure 1b), this approach abandons the use of labels and introduces the concept of pairwise similarity (the probability denoted by s_{ij} that the input x_i and x_j belong to the same class) for training. The binary classifier will assist the multi-class classifier in learning pairwise similarity to produce a discriminative model that predicts pairwise information.

Also, the paper develops different processes for the training of different learning paradigms. (You can find them in Figure 3 of the paper)

[1] https://github.com/COMP6248-Reproducability-Challenge/MULTI-CLASS_CLASSIFICATION_WITHOUT_MULTI-CLASS_LABELS



(a) Binary classifier ensemble (b) Meta classification learning (c) Geographical representation

Figure 1: Problem reduction schemes for multi-class classification. X_i represents the input data, Y_i represents the class label, S_{ij} represents the pairwise similarity between i and j, and θ is the learning parameters (Hsu et al., 2019)

2.2 Loss function

As shown in Figure 1c, the paper used a graphical model to identify the observed and unobserved information of the problem and accordingly defined the likelihood function $\mathcal{L}(\theta; \boldsymbol{X}; \boldsymbol{Y}, \boldsymbol{S})$.

Finally (Hsu et al., 2019) came up with the loss function L_{meta}. Note that in later charpters, we use MCL(Meta Classification Likelihood) as the learning criteria because of the likelihood nature of L_{meta}. ($\boldsymbol{X} = \{X_1, ...X_n\}$,$\boldsymbol{Y} = \{Y_1, ...Y_n\}$ and $\boldsymbol{S} = \{S_{ij}\}_{1 \leq i,j \leq n}$ represent samples, labels and pairwise similarities respectively and θ is the learning parameters)

$$\mathcal{L}(\theta; \boldsymbol{X}; \boldsymbol{Y}, \boldsymbol{S}) = P(\boldsymbol{S}|\boldsymbol{Y})P(\boldsymbol{Y}|\boldsymbol{X}; \theta)P(\boldsymbol{X})$$

$$L_{meta} = -\sum_{i,j} s_{ij} \log \hat{s}_{ij} + (1 - s_{ij}) log(1 - \hat{s}_{ij})$$

3 Experimental Methodology

We first tried to reproduce the results of the paper according to the model and parameters mentioned in the paper, and then we conducted additional experiments to evaluate the performance of MCL.

Experimental environment: All experiments are done on the ECS Yann GPU server, which has 4 1080ti GPUs, but the resources available to us are limited because it is a shared resource.

Therefore, we modified some parameters to increase the speed. Based on the implementation provided in the paper, we wrote some additional code to evaluate the classification performance of MCL on different datasets (MNIST, CIFAR10, CIFAR100 and Omniglot) and to compare it with networks using KCL (Kullback–Leibler divergence based contrastive loss) and CE (Cross-entropy) as loss functions.

3.1 Implementation and analysis

3.1.1 Supervise learning

Table 1 shows the experimental results we generated from the method proposed in this paper. The result we obtained were not much different from the original data (less than 5%). From the table, We can see that MCL usually yields a training effect similar to that of cross-entropy, and sometimes this model even gets a higher accuracy. This proves that the method of using similarity for classification is effective and well perform. Compared with using Kullback–Leibler divergence based contrastive loss(KCL)(Hsu & Kira, 2016) (Yen-Chang Hsu & Kira, 2016) which also use pairwise similarity, MCL has stronger generalization performance, and it can adapt to more state of the arts (at least it works better with ResNet than KCL). When it comes to CIFAR10, ResNet with KCL gives a poor classification result, the classification error rate has reached more than 70%. On a more complex data set (CIFAR100), MCL can get a relatively good classification performance (64.3% accuracy), which proves the feasibility of MCL on a more complex classification.

DATASET	CLASS	NETWORK	CE	KCL	MCL
MNIST	10	LeNet	0.58%	0.53%	0.60%
CIFAR10	10	LeNet	14.80%	15.42%	15.10%
		VGG8	10.42%	10.34%	9.99%
		VGG11	8.94%	68.30%	9.60%
		VGG16	7.59%	59.01%	8.22%
		ResNet18	6.52%	80.40%	6.56%
		ResNet34	6.55%	77.75%	6.28%
		ResNet50	5.98%	80.02%	5.78%
		ResNet101	5.97%	81.23%	5.45%
CIFAR100	10	VGG8	35.03%	42.00%	35.00%

Table 1: Experimental results on the error rate of supervised learning

There are two results (VGG11 with KCL and VGG16 with KCL) that we obtained are far from the results obtained by the author, which we believe is caused by different initialization of the parameters. We have tried that several times, the result we got best is about 15% loss rate and about 69% loss rate in the worst case.

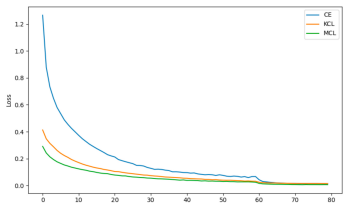


Figure 2: Running Loss of VGG8 with different Loss function(80 epoch) on MNIST

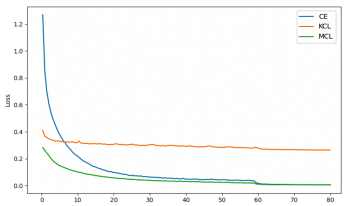


Figure 3: Running Loss of VGG11 with different loss function (80 epoch) on MNIST

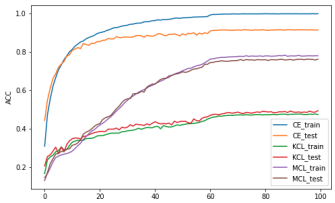


Figure 4: Training accuracy and testing accuracy of ResNet34 on MNIST (SGD)

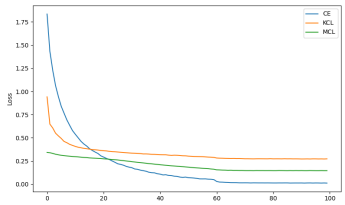


Figure 5: Running loss of ResNet34 with different loss function (100 epoch) on MNIST (SGD)

Figure 2 and 3 show that MCL converges faster than KCL and CE (Cross-entropy), and reach a loss of about 0.3 after the first epoch is completed (one epoch has 100 iterations). We have tried two VGG (VGG8 and VGG11) networks on the same data-set (MNIST). However, the performance of KCL fluctuates. When using VGG11, the loss of KCL stops decreasing when it reaches about 0.3, while CE and MCL can converge well to a loss value close to 0. We believe that the stability and generalisation performance of the MCL exceeds that of the KCL and CE.

When we trained with ResNet on MNIST using the Adam optimiser, the accuracy of the models using the three different loss functions was very different as Table 1 shows. We speculated whether this was due to a different optimiser, so we tried changing the optimiser to SGD and obtained the results shown in Figures 4 and 5. What disappointed us is that the accuracy of CE and MCL all have different degrees of decline, and the accuracy of KCL have increased. The accuracy rate of the CE test set dropped from 94% to 89%, the accuracy rate of MCL dropped from 93.7% to 76%, but the accuracy rate of KCL increased by about 20% (from 24% to 42%). We believe the reason for the lower accuracy of CE and MCL is that SGD is more likely to trap a model in a local minimum than Adam, so the model cannot converge to the optimal solution. In contrast, the original accuracy of KCL is not high (Adam) and due to the randomness of stochastic gradient descent, it happens to get a better result, despite this, the accuracy is still low.

3.1.2 UNSUPERVISED LEARNING

Method	Acc	Acc(K=100)	NMI	NMI(k=100)
MCL(Adan)	79.5%	75.9%	0.886	0.879
MCL(SGD)	18.5%	17.4%	0.354	0.331

Table 2: Unsupervised Learning Results on Omniglot

In the unsupervised learning scenario, we divided the 50 different letters in the Omniglot data-set into a background set and an evaluation set (30 and 20). And then learn the similarity function on the background set, and applies it to the cross-task transfer learning on the 20 evaluation sets. The results are shown in Table 2.

In the original text, Adam was used as the optimizer for training. When k=10, the result of the original paper got 83% accuracy and 0.897 NMI score, and got 80.2% acc and 0.893 NMI score when k=100. The results we obtain are slightly lower than the results presented by the paper, which may be caused by the difference in random initialization and our simplification of the model (we increased the learning rate to speed up the training process, and reduced the number of epochs).

Besides, we tried to use SGD as the optimizer, but the result is relatively poor. From Table 2, the accuracy we obtained was only 20% of the accuracy got from Adam, and the NMI score is only 40% of the result of Adam. We supposed this may because of the complicated data distribution of the Omniglot data set, which leads to a poor loss landscape and bad performance of SGD.

3.1.3 REPRODUCIBILITY ANALYSIS

We have successfully reproduced all the experiments of the original paper on supervised learning and the experiments on unsupervised learning. In terms of computational power, the university's servers provide us with support, but the sharing of resources made the time spent on the model training very high (at least 10 hours for one experiment). The most time-consuming part of the development process was to understand the paper and the code before making any changes and tuning.

The results are slightly different from those in the original paper, but still compatible. We believe that the reason why the results are reproducible is that the code is open source, the models used are all state of the arts, and the datasets are very common. However, we believe that the slight discrepancy from the results in the paper is not only due to the random initialisation of the parameters but also to the fact that the "tricks" (techniques that are correct in principle but not described in the paper) used in training or in processing the data to achieve certain results in the original paper are not fully disclosed.

4 CONCLUSION

We have reproduced and evaluated the novel strategy proposed by (Hsu et al., 2019) on supervised and unsupervised scenarios of multi-class classification. The results show that it is feasible to solve the classification problem through meta-problems with a suitable optimizer, and usually can obtain a classification effect similar to the result of using CE (Cross-entropy). Besides, we found that MCL has a stronger generalization performance than KCL.

REFERENCES

Yen-Chang Hsu and Zsolt Kira. Neural network-based clustering using pairwise constraints. *arXiv preprint arXiv:1511.06321*, 2016.

Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. *arXiv preprint arXiv:1901.00544*, 2019.

Zhaoyang Lv Yen-Chang Hsu and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. *arXiv preprint arXiv:1711.10125*, 2016.