To Shuffle or Not To Shuffle: Mini-Batch Shuffling Strategies for Multi-class Imbalanced Classification

Yuwei Mao, Vishu Gupta, Kewei Wang, Wei-keng Liao, Alok Choudhary, Ankit Agrawal
Department of Electrical and Computer Engineering
Northwestern University
Evanston, USA
{yuweimao2019,vishugupta2020,keweiwang2019}@u.northwestern.edu, {wkliao,choudhar,ankitag}@eecs.northwestern.edu

***Abstract*—Mini-batch shuffling is important for the deep learning training process. Most people use the random shuffling method, which aims to produce a random permutation of the training dataset in every epoch. In this study, we explore mini-batch shuffling for multi-class imbalanced data classification by investigating several shuffling strategies. We find that different order of input data can significantly affect the results of deep learning models. The results show that our proposed strategies can improve the accuracy by around 2%, demonstrating that higher diversity and lower imbalance ratio in each mini-batch can lead to better results.**

***Index Terms*—neural networks, shuffling, imbalanced classification, deep learning**

I. Introduction

Deep learning (DL) methods have achieved great success in recent years, owing to their impressive capacity to learn directly from raw data [1, 2, 3]. The ability of DL models heavily depends not only on the architectures and delicately-designed algorithms but also on the training dataset as well as the data ordering. Usually, people shuffle the training dataset randomly at the beginning of an epoch and divide it into mini-batches with different samples during training. Shuffling with mini-batches can help training converge faster and be more stable, preventing the model from learning the order of the training dataset. However, random shuffling may lead to different results across multiple deep learning experiments even when the architectures and parameters are identical, which can lead to difficulty reproducing the results. We investigate this problem in this study and explore some new mini-batch shuffling strategies for deep learning model training.

The classification problem is one of the most popular tasks in machine learning. Over the years, the performance of classification models has improved on high-quality synthetic datasets, such as CIFAR [4], ImageNet [5] and MS-COCO [6]. However, in many fields such as biology [7], materials science [8], manufacturing [9], social media [10], and others [11], data often exhibits class-imbalanced distributions. The imbalanced classification problems have attracted a lot of attention in the deep learning community [12]. In imbalanced data, the class that has more samples is the majority class and the class that has fewer samples is the minority class. Training deep learning models on imbalanced data is more challenging because the models may perform with bias towards the majority class and perform poorly on the minority class. Balancing the data distribution is usually used to solve this challenge when training the model. Resampling methods are commonly used for this purpose, which resample the training set to balance the dataset. There are two main resampling methods, oversampling and undersampling [13, 14, 15, 16]. Oversampling typically duplicates samples from the minority class, but it can cause over-fitting and increase the training time. Undersampling randomly removes samples from the majority class, however, it may discard potentially useful data.

Compared to binary imbalanced classification, multi-class imbalanced classification tasks are more challenging [17] and have received significant research attention in recent years [18]. Most existing solutions for multi-class imbalanced classification use class decomposition schemes to handle multi-class and work with two-class imbalance techniques to handle each imbalanced binary subtask. However, it can aggravate imbalanced distributions [19], and combining results from classifiers learned from different subproblems can cause potential classification errors [20].

In this work, we explore a few mini-batch shuffling strategies for the original data itself instead of oversampling or undersampling methods for multi-class imbalanced data classification to avoid over-fitting problems while making full use of all available data. First, we explore random shuffling for multi-class imbalanced data by running multiple experiments with the same parameters. Then, we propose some new mini-batch shuffling strategies for multi-class imbalanced data.

II. Methods

The random shuffling method randomly shuffles the training dataset and divides it into mini-batches at the beginning of each epoch. By running multiple experiments with the same initial weights and parameter settings but different data order, we know that random shuffling for multi-class imbalanced data classification gives different results across multiple experiments. The intuition here is that each mini-batch may have a different data distribution for multi-class imbalanced data across different experiments, especially when there are many classes and the imbalance ratio is high.

In this work, we propose three shuffling strategies that consider both class and imbalance ratios for mini-batch shuffling. The proposed strategies randomly shuffle the training data, then select samples for each mini-batch according to different

979-8-3503-2028-2/22/$31.00 ©2022 IEEE
DOI 10.1109/CSCI58124.2022.00157

criteria. We run every shuffling strategy multiple times with the same initialization and parameter settings for evaluation and comparison. The imbalanced dataset we used is constructed from CIFAR-10. We use 5 majority classes and 5 minority classes with a ratio of 10:1 to construct the dataset. Thus, the training dataset has 27,500 images (5 majority classes of 5,000 images each, and 5 minority classes of 500 images each) and 5,500 validation images (5 majority classes of 1,000 images each, and 5 minority classes of 100 images each). The specific construction method will be introduced in the next section.

The first strategy is called ”class with imbalance”. The criterion of selecting samples for each mini-batch here is to make sure each mini-batch includes all the 10 classes and the ratio between majority classes and minority classes is 10:1, i.e., $\frac{C_i^{maj}}{C_j^{min}} = 10$, where C_i^{maj} is the number of samples in majority class i and C_j^{min} is the number of samples in minority class j. When the batch size is N, the number of samples in majority class $C_i^{maj} = \frac{2N}{11}$, the number of samples in minority class $C_j^{min} = \frac{N}{55}$. Compared to the random shuffling method, each mini-batch here has the same data distribution as that of the whole dataset.

The second strategy is called ”class with balance”. Inspired by the resampling strategies, we assume that the balanced data may increase the accuracy of deep learning models. Thus, we design a strategy to construct balanced data in each mini-batch. The initial mini-batches in every epoch include 10 classes with the same number of samples. After all minority classes are assigned to initial mini-batches, the remaining mini-batches only include 5 majority classes with the same number of samples.

The third strategy is called ”5 classes with balance”. Similar to the second strategy, this strategy is to construct balanced data in each mini-batch. Instead of each mini-batch having 10 classes for the initial mini-batches in every epoch, this strategy tries to make sure that each mini-batch includes 5 classes with the same number of samples. Here the initial mini-batches include 5 minority classes. After all minority classes are assigned to initial mini-batches, the remaining mini-batches include 5 majority classes.

III. Experiments and Results

A. Dataset

We created an imbalanced dataset using CIFAR-10. The original dataset contains 50,000 training images and 10,000 validation images of size 32 × 32 with 10 classes. Each class has 5,000 training images and 1,000 validation images. We randomly select 5 classes as majority classes and the other 5 classes as minority classes. The majority classes have the same number of images as before, while the minority classes only have $1/10$ of the original data. Our new class-imbalanced dataset from the CIFAR-10 dataset thus contains 27,500 training images and 5,500 validation images with 10 classes. Each majority class has 5,000 training images and 1,000 validation images, while each minority class has 500 training images and 100 validation images.

B. Experimental Settings

We use a simple convolutional neural network (CNN) [21] architecture here. For each strategy, 20 runs of model training are performed with the same weights initialization, the number of epochs, and network hyperparameters, but different random shuffling seeds. The CNN model we used as shown in Figure 1. In this model, the first and second convolutional layers have 16 and 32 3×3 kernels, respectively, followed by a 2×2 max pooling layer. The third convolutional layer has 64 3×3 kernels, which is followed by two fully connected layers where the first and second layers contain 1,024 and 10 neurons, respectively. Cross entropy loss is used as the loss function with Adam optimizer and setting a learning rate of 0.0001. The batch size is 512. The number of epochs is 100. We use the best validation accuracy to compare the results. We implemented all experiments in PyTorch with an NVIDIA Quadro RTX 8000 GPU. The PyTorch shuffling seed is set as 34 for weights initialization. In k-th run, the random shuffling seed for the training set for i-th epoch is $k \times 20 + i$. The code for data generation and all methods are available at https://github.com/MaoYuwei/batch_shuffling.

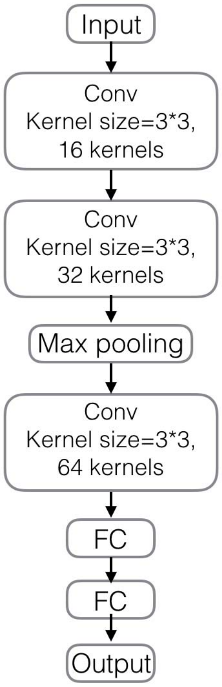
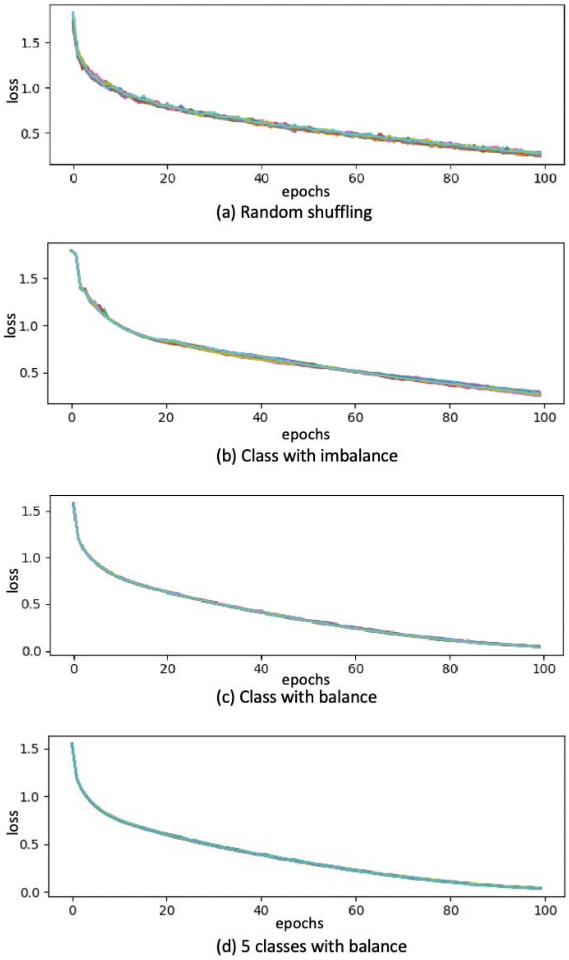


Fig. 1. The architecture of the CNN model.

C. Results

Figure 2 shows the training loss curves of 20 runs of different shuffling strategies. Figure 3 shows the validation loss curves of 20 runs of different shuffling strategies. We can see the variation in training loss and validation loss across multiple runs. Figure 2(a) shows that the training loss of different runs are different even though the parameters and architectures are the same when training the deep learning models. Similar results can be observed in 3(a), where the validation loss of different runs is different. This observation demonstrates that random shuffling will lead to different results across multiple deep-learning experiments.



(a) Random shuffling

(b) Class with imbalance

(c) Class with balance

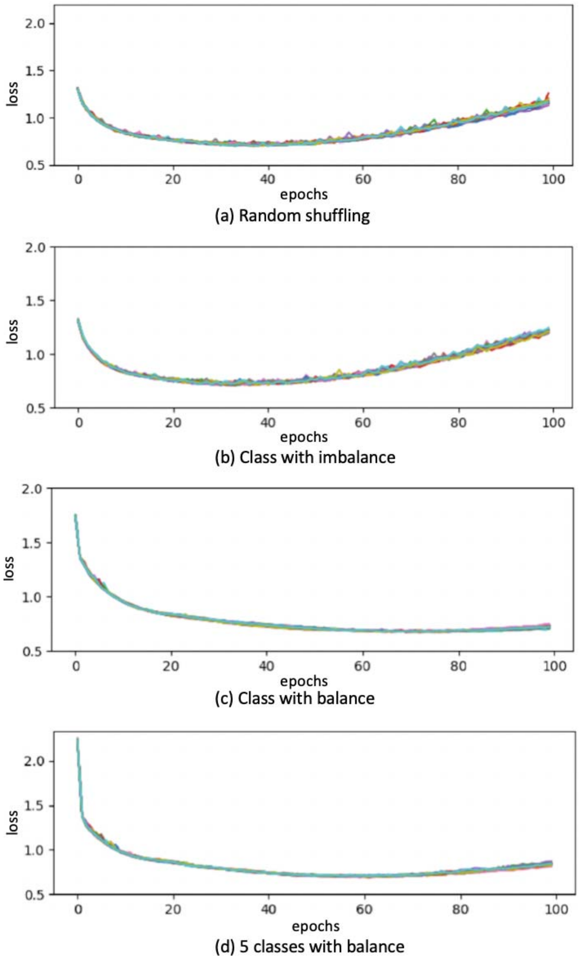
(d) 5 classes with balance

Fig. 2. Training loss of different shuffling strategies.

TABLE I
RESULTS OF DIFFERENT SHUFFLING STRATEGIES

Strategies	mean(%)	stddev(%)
random shuffling	78.0118	**0.1301**
class with imbalance	78.4264	0.1355
class with balance	79.4373	0.1870
5 classes with balance	**79.6109**	0.1730

The mean value and standard deviation of the accuracy results are shown in Table I. It shows that the mean accuracy is better when we consider class for mini-batch shuffling rather than simple random shuffling on multi-class imbalanced data. The mean accuracy of the proposed strategies is about 2% better than the random shuffling method, suggesting that diversity within each mini-batch is important for better learning. Random shuffling in a highly unbalanced scenario may not guarantee enough diversity within each mini-batch. The proposed strategies increase the diversity in each mini-batch, which means the samples in each mini-batch are less similar or clustered.



(a) Random shuffling

(b) Class with imbalance

(c) Class with balance

(d) 5 classes with balance

Fig. 3. Validation loss of different shuffling strategies.

We also observe that the results of mini-batch shuffling strategies with balance are slightly better than mini-batch shuffling strategies with imbalance. And Figure 3(a)(b) show that overfitting occurs after 40 epochs when we use random shuffling and class with imbalance strategies. Figure 3(c)(d) show that mini-bath shuffling strategies with balance do not cause obvious overfitting with the same epoch number. We believe it might be due to the fact that the classification for balanced data is easier than for imbalanced data. If each mini-batch has balanced data, the model apparently learns better. The best mean accuracy is obtained with ”5 classes with balance”, which is 79.6109%.

IV. Conclusion

In this study, we explore some shuffling strategies for multi-class imbalance data classification. The results show that multiple experiments get different results even with the same parameters, and reveal valuable insights into the impact of shuffling on model accuracy and the importance of diversity and data balance within each mini-batch. Preliminary results also suggest that shuffling considering both class and imbalance ratio may improve the results compared to random shuffling. In the future, we will expand our exploration of mini-batch shuffling strategies to different architectures and more datasets, such as different ratio imbalanced data, multi-labeled data, and long-tailed class data.

Acknowledgment

This work is supported in part by the following grants: NIST award 70NANB19H005; DOE awards DE-SC0019358, DE-SC0021399; NSF award CMMI-2053929.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[2] Yuwei Mao, Zijiang Yang, Dipendra Jha, Arindam Paul, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Generative adversarial networks and mixture density networks-based inverse modeling for microstructural materials design. *Integrating Materials and Manufacturing Innovation*, pages 1–11, 2022.

[3] Vishu Gupta, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Brnet: Branched residual network for fast and accurate predictive modeling of materials properties. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 343–351. SIAM, 2022.

[4] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11), 2008.

[5] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[7] Tian-Yu Liu. Easyensemble and feature selection for imbalance data sets. In *2009 international joint conference on bioinformatics, systems biology and intelligent computing*, pages 517–520. IEEE, 2009.

[8] Ankit Agrawal and Alok Choudhary. Deep materials informatics: Applications of deep learning in materials science. *MRS Communications*, 9(3):779–792, 2019.

[9] Yuwei Mao, Hui Lin, Christina Xuan Yu, Roger Frye, Darren Beckett, Kevin Anderson, Lars Jacquemetton, Fred Carter, Zhangyuan Gao, Wei-keng Liao, et al. A deep learning framework for layer-wise porosity prediction in metal powder bed fusion using thermal signatures. *Journal of Intelligent Manufacturing*, pages 1–15, 2022.

[10] Peng Liu, Wei Chen, Gaoyan Ou, Tengjiao Wang, Dongqing Yang, and Kai Lei. Sarcasm detection in social media based on imbalanced classification. In *International Conference on Web-Age Information Management*, pages 459–471. Springer, 2014.

[11] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, 73:220–239, 2017.

[12] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019.

[13] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[14] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[15] Yusheng Xie, Zhengzhang Chen, Ankit Agrawal, Alok Choudhary, and Lu Liu. Random walk-based graphical sampling in unbalanced heterogeneous bipartite social graphs. In *Proceedings of 22nd ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1473–1476, 2013.

[16] Reda Al-Bahrani, Dipendra Jha, Qiao Kang, Sunwoo Lee, Zijiang Yang, W. Liao, Ankit Agrawal, and Alok Choudhary. Sigrnn: Synthetic minority instances generation in imbalanced datasets using a recurrent neural network. In *Proceedings of International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 349–356, 2021.

[17] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1):63–77, 2005.

[18] Shuo Wang and Xin Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1119–1130, 2012.

[19] Aik Choon Tan, David Gilbert, and Yves Deville. Multi-class protein fold classification using a new ensemble machine learning approach. *Genome Informatics*, 14:206–217, 2003.

[20] Rong Jin and Jian Zhang. Multi-class learning by smoothed boosting. *Machine learning*, 67(3), 2007.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 2017.