Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 242 (2024) 1112–1117

Procedia Computer Science

www.elsevier.com/locate/procedia

11th International Conference on Information Technology and Quantitative Management (ITQM 2024)

Latest Technologies on Dataset Distillation: A Survey

Muyang Li[a,b,c], Yi Qu[a,b,d], Yong Shi[a,b,e,*]

[a]*Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China*
[b]*Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China*
[c]*School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing,100190,China*
[d]*School of Economics and Management, University of Chinese Academy of Sciences, Beijing,100190,China*
[e]*College of Information Science and Technology, University of Nebraska at Omaha, Omaha, NE 68182, USA*

Abstract

Dataset distillation refers to the process of constructing a smaller dataset based on a larger dataset, so that the training model with the smaller dataset can obtain similar results to the larger dataset. In this paper, we first briefly review the basic methods and some recent advances in this field from the perspectives of meta-learning framework and data matching framework, including the backpropagation through time(BPTT), kernel ridge regression(KRR), decoupled techniques, gradient matching(DC), trajectory matching(MTT) and distribution matching(DM). The results of the plug-and-play approach in recent years are then presented. Finally, we discuss possible future developments from the perspectives of data distillation algorithms on large datasets, the relationship between compress ratio and information contained, more tasks and data modalities, and the use of generative models to improve the distillation results of datasets.

Keywords: Dataset Distillation; Survey; Meta-learning framework; Data matching framework

1. Introduction

During past decades, deep learning has made notable success in areas such as computer vision (CV) [11, 24], natural language processing (NLP) [6], anomaly detection [22], graph neural network (GNN) [26] and time series forecasting [25]. The success of deep learning is largely due to the use of large amounts of high-quality training data. While acknowledging the success of deep learning, it is necessary to recognize its dependence on vast quantities of high-quality training data. The sheer size of datasets makes data storage, distribution, and maintenance extremely costly. In deep learning, training a deep neural network typically requires training thousands of epochs on the entire

* Corresponding author. Yong Shi
E-mail address: yshi@ucas.ac.cn

1877-0509
10.1016/j.procs.2024.08.188

dataset. In particular, certain computational tasks such as grid search and Neural Architecture Search (NAS) [23] necessitate the large number of repeating training procedures. This implies higher time costs for executing such tasks on larger datasets.

Considerable work has been done to reduce the time cost of deep learning tasks. Examples include continuous advancements in GPU computing power, the development of specialized hardware like Neural Processing Unit (NPU) [2] and Tensor Processing Unit (TPU) [14] for tensor computations, the invention of highly parallel network structures such as transformers [27], and applying knowledge distillation methods to transfer the knowledge from the more complex model to the lightweight model [13]. However, these approaches mainly concentrate on enhancing computational power and algorithmic efficiency, while potential optimizations at the data level still remain to be explored.

Consequently, we can rise such question naturally: is it possible to construct a small dataset that, when taken as training data, will be able to obtain a network with comparable performance to the network trained on a larger original dataset?

Then we provide a mathematical expression of this problem.

For given dataset $\mathcal{T}$, construct another dataset $\mathcal{S}$ such that

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[l(f_{\mathtt{alg}(\mathcal{S})}(\mathbf{x}),y)] \approx \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[l(f_{\mathtt{alg}(\mathcal{T})}(\mathbf{x}),y)] \tag{1}$$

Here l here refers to the loss function. f_θ denotes the neural network with parameter θ. $\mathtt{alg}(\mathcal{S})$ and $\mathtt{alg}(\mathcal{T})$ denotes the parameters of neural network obtained by training on dataset $\mathcal{S}$ and $\mathcal{T}$, respectively. $\mathcal{R} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[l(f_\theta(\mathbf{x}),y)]$ is the expected risk of model f_θ on distribution $\mathcal{D}$. In practical applications, due to the unknown of distribution $\mathcal{D}$, we often employ the empirical loss $\mathcal{R}_\mathcal{T} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{T}}[l(f_\theta(\mathbf{x}),y)]$ computed on $\mathcal{T}$ as a surrogate for the expected risk.

Regarding this particular problem, recent solutions have been categorized into two approaches: coreset selection method [9] and dataset distillation method. The coreset selection approach refers to the process of identifying a subset of data from the original dataset $\mathcal{T}$ that significantly influences the training process of the model to constitute the set $\mathcal{S}$. A defining feature of this process is that the resulting set $\mathcal{S}$ is invariably a subset of $\mathcal{T}$. For example, in a support vector machine, the process of solving a support vector is a process of selecting a core set. Because the final decision hyperplane is only influenced by the support vectors. The dataset distillation approach places no restrictions on the methods used to produce the set $\mathcal{S}$, with the resultant set not necessarily being a subset of $\mathcal{T}$. This methodology is also called Dataset Condensation. The purpose of this paper is to provide a comprehensive review of some significant methods of dataset distillation, as well as the methodologies that have been newly introduced between the years 2022 and 2024.

Based on the optimization objects within the algorithms, dataset distillation methods are categorized into two distinct categories: meta-learning frameworks, data matching frameworks and factorized dataset distillation [16]. This paper mainly focus on meta-learning frameworks and data matching frameworks.

The rest of this paper will be arranged as follows: Section 2 and 3 will review the Meta-learning Framework and Data Matching Framework, respectively. Section 4 will review some plug-and-play methods of dataset distillation. Section 5 will discuss some challenges and future direction of dataset distillation.

2. Meta-learning framework

Meta-learning Framework consider the set $\mathcal{S}$ as a learnable parameter of a deep learning problem. $\mathcal{S}$ represents the outcome of an optimization problem: Determine dataset $\mathcal{S}$ such that, for a given neural network architecture, the resultant training on this dataset can minimize the expected risk.

The meta-learning framework can be written as:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathcal{R}_\mathcal{T}(\mathtt{alg}(\mathcal{S})) \tag{2}$$

subject to

$$\mathtt{alg}(\mathcal{S}) = \arg\min_{\theta} \mathcal{R}_{\mathcal{S}}(\theta) \tag{3}$$

Equ.2 is called the outer optimization and Equ.3 is called the inner optimization of the meta-learning framework [16].

Classified by different architecture inner loop, the meta-learning framework can be divided into three categories: the backpropagation through time (BPTT), the kernel ridge regression (KRR) and the decoupled techniques.

2.1. Backpropagation through time (BPTT)

The BPTT framework uses neural network as the model for inner optimization problem. Wang et al. employed this approach in the article that first introduced the concept of Data Distillation [29]. However, due to the utilization of deep learning methodologies for both the inner and outer optimization problems within this method, the BPTT process is typically time-consuming. To improve the efficiency, feng et al. imported the random truncation and proposed the RaT-BPTT method, which can accelerate the optimization process while maintaining long term dependencies [8].

2.2. Kernel ridge regression (KRR)

The KRR framework uses kernel ridge regression model for the inner optimization problem [21]. The advantage of employing a KRR model in the inner optimization is that the KRR model possesses a closed-form solution. Consequently, this can significantly reduce the computational time required for the inner optimization procedure while maintaining a sufficiently good performance of the inner model.

Maalouf et al. applied a theoretical analysis on KRR framework by transforming ridge regression in random Fourier features(RFF) space [20]. Then they proved the existence of distilled datasets under KRR framework and the solution of KRR with distilled datasets is an approximation of that with the original dataset. They also provided an error bound of distilled dataset.

2.3. Decoupled technique

Researches has showed that the knowledge can be stored in pre-trained models [19]. The decoupled techniques tries to avoid the training progress of the inner optimization by using a pre-trained model in inner optimization. The application of pre-trained models can reduce the time required for the inner-level optimization problem within the meta-learning framework to an order of O(1) [30].

3. Data matching framework

Data matching framework is another framework in dataset distillation. It synthesizes dataset $\mathcal{S}$ by matching certain indicators between the two datasets $\mathcal{S}$ and $\mathcal{T}$.

3.1. Gradient matching(DC)

The idea of the gradient matching approach is that if two neural networks share the same architecture, initial parameters, training strategy and gradients at each step of training, differing only in the datasets they use, then the two datasets can be considered to be equivalent [36].

The optimization process of one model can be regarded as the movement of all parameters of the model in the parameter space. Under this view, the gradient matching method can be thought of as minimizing the difference between the first order derivative of the respective trajectories of the two models. Consequently, a model trained on a synthetic dataset $\mathcal{S}$ can achieve performance comparable to that of a model trained on the original dataset $\mathcal{T}$.

3.2. Trajectory matching(MTT)

Similar to the gradient matching method, the trajectory matching also trains two models with same architecture, initial parameters and training strategy. The difference is trajectory matching directly uses the distance between the parameters of the two models within the parameter space as the loss function [1]. One of the major drawbacks of the trajectory matching method is its high memory cost. In [4], Cui et al. reduced the memory requirements of the trajectory matching method to a constant level by reformulating the backpropagation expression of the trajectory matching method [10]. This reformulation condensed the number of computational graphs to a single one. Then, the MTT was firstly applied into the ImageNet-1K dataset with higher image per class (IPC), which is often the main reason for consuming memory. Guo et al. noted that neural networks learn patterns of different complexity at different stages of training, as well as the relationship between different patterns and the size of the synthetic dataset. Based on this, Guo et al. proposed one lossless dataset distillation method within MTT framework.

Similar to the gradient matching, the trajectory matching method can be thought of as directly minimizing the difference between the model parameters of the respective trajectories of the two models in the parameters space.

3.3. Distribution matching(DM)

Contrary to the aforementioned two matching methods, the distribution matching approach aims to perform the task of dataset distillation by aligning the distribution of synthesized dataset $\mathcal{S}$ and that of the original dataset $\mathcal{T}$.

Zhao and Bilen synthesized a synthetic dataset $\mathcal{S}$ by minimizing the distance between the mean features of elements in set $\mathcal{S}$ and set $\mathcal{T}$ [35]. They select a neural network, which is randomly initialized, as a feature extractor for the elements within both dataset, and only considered the output of the neural network. In [28], the authors employed a feature alignment method, which necessitates that when synthetic data and original data are input into the same neural network, the outputs at each layer of the network should be consistent and proposed the CAFE method.

In [37], the author pointed two shortcomings of the naive distributing matching method: the imbalanced feature number between original dataset and synthetic dataset and the inappropriate selection of loss function in distribution matching. To address these two shortcomings, the author has employed a technique that incorporates partitioning and expansion augmentation to rectify the imbalance in the number of features between the synthetic and original datasets. Additionally, an approach known as Enriched Model Sampling has been utilized to enhance the accuracy of feature extraction. Furthermore, the application of a cross-entropy loss function has improved the precision in the feature matching process.

In the original DM method proposed by Zhao and Bilen, only the means of the features from the synthetic and original datasets were aligned. In contrast, [32] embedded the data distribution into a Reproducing Kernel Hilbert Space (RKHS), thereby aligning all orders of moments between the distributions of real and synthetic images, which leads to the generation of the synthetic dataset $\mathcal{S}$.

Deng et al. observed that in the native DM method, the feature distribution of synthetic samples in the same class is relatively dispersed. To address this issue, the authors have introduced two constraints: the class centralization constraint and the covariance matching constraint [5]. These constraints significantly enhance the performance of the synthetic dataset in classification problem. Furthermore, through experimentation, the authors have demonstrated that the integration of proposed method with various mainstream DM approaches consistently improve performance.

4. Plug-and-play method

Duan et al. has noted that current methodologies for dataset distillation are associated with high time and space complexity, while the synthesized data also should have high information density. The majority of dataset distillation methods can only address one or two of the three aforementioned challenges. In order to comprehensively tackle all three issues, Duan et al. has proposed a novel approach LatentDD [7]. This method effectively mitigates the consumption of time and space by representing both the original and synthesized data as latent vectors, thereby achieving a significant reduction in computational demands. Furthermore, it demonstrates a high degree of compatibility with various mainstream dataset distillation methods.

By applying Logit-Based Prediction Error (LBPE) score and balanced construction as dataset pruning rule, He et al. constructed a smaller distillation dataset based on the distilled dataset again [12]. This work not only improved the performance of the model under the same compression ratio, but also provided the flexibility to adjust the size of the dataset to adapt different computational constraints.

For the matching framework, the current strategy for selecting original images is limited to random sampling. This strategy ignores the evenness and diversity of the selected sample distribution, which can lead to biased matching results. Liu et al. selects representative raw images for matching, which effectively reduces the number of iterations of the algorithm without affecting performance [17, 18].

5. Conclusion and discussion

In traditional dataset distillation methodologies, the datasets commonly utilized are relatively smaller ones, such as MNIST and CIFAR-10 [29]. However, recent advancements in the optimization of memory and time efficiency have led to a shift towards larger datasets, including TinyImageNet and ImageNet-1k. We believe that low memory consumption and efficient dataset distillation methods applied to large datasets will become mainstream in this field.

Image per class (IPC) is an important parameter to describe the dataset, which describes the number of data points in each class in a categorical dataset, for example, IPC=5000 in the cifar-10 training dataset [15]. This concept has also been introduced into dataset distillation to describe the compression ratio of dataset distillation algorithms. Yu pointed out that the performance of data distillation only exceeds that of the selection-based method in the case of IPC $<$ 200 [31]. Considering that extremely low compression ratio is not the ultimate goal of data distillation, we propose whether there is a data distillation method that can improve the performance of the compressed dataset at the expense of compression ratio. We believe that this problem involves the field of multi-criteria optimization. In addition, this phenomenon suggests that most of the information contained in the dataset can be stored in a very small amount of data, which means that the information in the dataset can be considered to be composed of ”critical information” and ”redundant information”, and how to strictly characterize these two is also a question to be studied. The study of this problem helps us to condense the dataset by removing redundancies without losing valid information.

The majority of current research in dataset distillation are based on classification tasks and utilizing these as the benchmark for evaluation [3]. Although there exist studies that suggest the potential for the extension of dataset distillation methods to domains such as text data, graph data, and medical images [33], how to design distillation algorithms for data in other modalities and other computer vision problem datasets still needs to be further studied.

One representation of dataset distillation is to represent the information in the original dataset with a much smaller amount of data. So we consider a generative model that can generate training data equivalent to the original dataset, but with parameters much less than the amount of data in the original dataset, and consider it as a generalized dataset distillation problem. Guided by this idea, Zhao and Bilen built a GAN-based data generator and applied it to the distillation cifar-10 model [34]. Based on this fact, we ask that whether it is possible to build a generative model based on the original dataset so that the data generated can be used for model training to achieve comparable or even better performance than the original dataset?

Acknowledgements

This work has been funded by Key Projects of National Natural Science Foundation of China (#71932008, #72231010). We express our sincere appreciation to Github project Awesome-Dataset-Distillation for its up-to-date summary of dataset distillation. We also would like to show our deepest gratitude to all the editors and reviewers for their comments.

References

[1] Cazenavette, G., Wang, T., Torralba, A., Efros, A.A., Zhu, J.Y., 2022. Dataset distillation by matching training trajectories, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4750–4759.

[2] Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y., Temam, O., 2014. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. ACM SIGARCH Computer Architecture News 42, 269–284.

[3] Cui, J., Wang, R., Si, S., Hsieh, C.J., 2022. Dc-bench: Dataset condensation benchmark. Advances in Neural Information Processing Systems 35, 810–822.
[4] Cui, J., Wang, R., Si, S., Hsieh, C.J., 2023. Scaling up dataset distillation to imagenet-1k with constant memory, in: International Conference on Machine Learning, PMLR. pp. 6565–6590.
[5] Deng, W., Li, W., Ding, T., Wang, L., Zhang, H., Huang, K., Huo, J., Gao, Y., 2024. Exploiting inter-sample and inter-feature relations in dataset distillation. arXiv preprint arXiv:2404.00563 .
[6] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .
[7] Duan, Y., Zhang, J., Zhang, L., 2023. Dataset distillation in latent space. arXiv preprint arXiv:2311.15547 .
[8] Feng, Y., Vedantam, S.R., Kempe, J., 2023. Embarrassingly simple dataset distillation, in: The Twelfth International Conference on Learning Representations.
[9] Guo, C., Zhao, B., Bai, Y., 2022. Deepcore: A comprehensive library for coreset selection in deep learning, in: International Conference on Database and Expert Systems Applications, Springer. pp. 181–195.
[10] Guo, Z., Wang, K., Cazenavette, G., Li, H., Zhang, K., You, Y., 2023. Towards lossless dataset distillation via difficulty-aligned trajectory matching. arXiv preprint arXiv:2310.05773 .
[11] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
[12] He, Y., Xiao, L., Zhou, J.T., 2024. You only condense once: Two rules for pruning condensed datasets. Advances in Neural Information Processing Systems 36.
[13] Hinton, G., Vinyals, O., Dean, J., 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 .
[14] Jouppi, N.P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al., 2017. In-datacenter performance analysis of a tensor processing unit, in: Proceedings of the 44th annual international symposium on computer architecture, pp. 1–12.
[15] Krizhevsky, A., Hinton, G., et al., 2009. Learning multiple layers of features from tiny images .
[16] Lei, S., Tao, D., 2023. A comprehensive survey of dataset distillation. IEEE Transactions on Pattern Analysis and Machine Intelligence .
[17] Liu, Y., Gu, J., Wang, K., Zhu, Z., Jiang, W., You, Y., 2023a. Dream: Efficient dataset distillation by representative matching, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17314–17324.
[18] Liu, Y., Gu, J., Wang, K., Zhu, Z., Zhang, K., Jiang, W., You, Y., 2023b. Dream+: Efficient dataset distillation by bidirectional representative matching. arXiv preprint arXiv:2310.15052 .
[19] Lu, Y., Chen, X., Zhang, Y., Gu, J., Zhang, T., Zhang, Y., Yang, X., Xuan, Q., Wang, K., You, Y., 2023. Can pre-trained models assist in dataset distillation? arXiv preprint arXiv:2310.03295 .
[20] Maalouf, A., Tukan, M., Loo, N., Hasani, R., Lechner, M., Rus, D., 2023. On the size and approximation error of distilled sets. arXiv preprint arXiv:2305.14113 .
[21] Nguyen, T., Chen, Z., Lee, J., 2020. Dataset meta-learning from kernel ridge-regression. arXiv preprint arXiv:2011.00050 .
[22] Pang, G., Shen, C., Cao, L., Hengel, A.V.D., 2021. Deep learning for anomaly detection: A review. ACM computing surveys (CSUR) 54, 1–38.
[23] Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X., Wang, X., 2021. A comprehensive survey of neural architecture search: Challenges and solutions. ACM Computing Surveys (CSUR) 54, 1–34.
[24] Shi, Y., Cui, L., Qi, Z., Meng, F., Chen, Z., 2016. Automatic road crack detection using random structured forests. IEEE Transactions on Intelligent Transportation Systems 17, 3434–3445.
[25] Shi, Y., Wang, Y., Qu, Y., Chen, Z., 2024a. Integrated gcn-lstm stock prices movement prediction based on knowledge-incorporated graphs construction. International Journal of Machine Learning and Cybernetics 15, 161–176.
[26] Shi, Y., Zheng, L., Quan, P., Niu, L., 2024b. Wasserstein distance regularized graph neural networks. Information Sciences , 120608.
[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.
[28] Wang, K., Zhao, B., Peng, X., Zhu, Z., Yang, S., Wang, S., Huang, G., Bilen, H., Wang, X., You, Y., 2022. Cafe: Learning to condense dataset by aligning features, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12196–12205.
[29] Wang, T., Zhu, J.Y., Torralba, A., Efros, A.A., 2018. Dataset distillation. arXiv preprint arXiv:1811.10959 .
[30] Yin, Z., Xing, E., Shen, Z., 2024. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. Advances in Neural Information Processing Systems 36.
[31] Yu, R., Liu, S., Wang, X., 2023. Dataset distillation: A comprehensive review. IEEE Transactions on Pattern Analysis and Machine Intelligence .
[32] Zhang, H., Li, S., Wang, P., Zeng, D., Ge, S., 2024. M3d: Dataset condensation by minimizing maximum mean discrepancy, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9314–9322.
[33] Zhao, B., 2023. Data-efficient neural network training with dataset condensation .
[34] Zhao, B., Bilen, H., 2022. Synthesizing informative training samples with gan. arXiv preprint arXiv:2204.07513 .
[35] Zhao, B., Bilen, H., 2023. Dataset condensation with distribution matching, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 6514–6523.
[36] Zhao, B., Mopuri, K.R., Bilen, H., 2020. Dataset condensation with gradient matching. arXiv preprint arXiv:2006.05929 .
[37] Zhao, G., Li, G., Qin, Y., Yu, Y., 2023. Improved distribution matching for dataset condensation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7856–7865.