UNIVERSITY OF
TORONTO

Department of Electrical & Computer Engineering

ECE1512 - Project B: Dataset Distillation

Author:	*Student Number:*
Yiyang (Fred) Shi	1001959589
Hsuan-Ling (Celene) Chen	1002322202

Date Due: December 12, 2022

Date Handed in: December 12, 2022

ECE1512 - Project B: Dataset Distillation

Yiyang (Fred) Shi, Hsuan-Ling (Celene) Chen

December 12, 2022

Abstract

1 Introduction

Unlike classical data compression, dataset distillation aims for a small synthetic dataset that still retains adequate task-related information so that models trained on it can generalize to unseen test data. Distilling algorithm must strike a delicate balance by heavily compressing information without completely obliterating the discriminative features.

1.1 Background

In order to reduce the high computing source challenge of deep learning model trained on the large-scaled datasets, dataset distillation (DD) was a new option introduced by Wang et al [1]. to reduce computation cost based model-space approach. The dataset distillation produces a synthetic dataset of small size which assures the same performance of the learned model compared to the model learned on the full dataset. Therefore can help to reduce the required memory and computation for a deep learning model.

1.2 Objective

This project aims to use tools and technologies to set up dataset distillation as a data compression technique. First part of project uses the prior dataset distillation with Gradient Matching as a data compression method on MNIST and CIFAR10; The second part of the project uses two state-of-the-art methods, Trajectory Matching and Distribution Matching to further explore the effect of dataset distillation framework on visual classification tasks.

2 Task 1

2.1 Task Overview

2.1.1 1a: Purpose of Dataset Condensation

This paper [2] proposes a training set synthesis technique for data-efficient learning, called Dataset Condensation, that learns to condense large dataset into a small set of informative synthetic samples for training deep neural networks from scratch. The small synthetic dataset should achieve comparable accuracy results to the large dataset regardless of network architecture used for training.

The purpose of using Data Distillation in the paper is to obtain comparable generalization performance by training on the condensed data and aims to find the optimum set of synthetic images such that the model trained on them minimizes the training loss over the original data.

2.1.2 1b: The advantages of the methodology over state-of-the-art

The authors model the network parameters as a function of the synthetic training data and learn them by minimizing the training loss over the original training data w.r.t. synthetic data. The method goes beyond relying on the limitations in heuristics that does not guarantee

any optimal solution for the downstream task and presence of representative samples, which is neither guaranteed. Unlike in the coreset methods, the synthesized data are directly optimized for the downstream task and thus the success of the method does not rely on the presence of representative samples. It focus on learning to synthesize informative samples that are optimized to train neural networks for downstream tasks and not limited to individual samples in the original dataset.

2.1.3 1c: Contributions compared to prior methods

Compare to Dataset Distillation [1], gradient matching method simplifies and decomposes the nested optimization problem, which is computationally expensive to solve, into multiple sub-problems. The method tries to match the parameters trained on synthetic data θ^S to the final parameters trained on original data θ^T as well as the optimization trajectory throughout the optimization process. Therefore, the rapid unrolling of recursive computation graph over multiple optimization steps over previous parameters is not needed. The optimization algorithm only need to ensure that θ^S is similar to θ^T at each iteration. The optimization process is significantly faster and more memory efficient.

2.1.4 1d: Metholodogies

The paper is based on the optimization problem proposed in (Wang et al., 2018) where it expresses the optimization problem as

$$S^* = \underset{S}{argmin} L^T(\theta^S(S)) \;\; subject\ to \;\; \theta^S(s) = \underset{\theta}{argmin} L^S(\theta))$$

Parameter matching method extends from the approach shown in equation 3 above and replace the inner loop $\theta^S(S)$ optimization with a parameter matching optimization between original dataset T and condensed dataset S. With compariable parameters θ^S and θ^T, the model should converge to a similar solution between the 2 datasets, therefore guarantee the performance of the condensed dataset. In addition, this optimization should work for a distribution of random initailization of parameters $theta_0$ and optimization goal can be reorganized into

$$\underset{S}{min} E_{\theta_0 \sim P_\theta}[D(\theta^S(\theta_0), \theta^T(\theta_0)] \;\; subject\ to \;\; \theta^S(s) = \underset{\theta}{argmin} L^S(\theta(\theta_0))$$

Extended from parameter matching, the curriculum gradient matching approach decomposes the nested optimization problem proposed parameter matching method into multiple sub-problems at each iteration.

$$\underset{S}{min} E_{\theta_0 \sim P_\theta}[\sum_{t=0}^{T-1} D(\theta_t^S, \theta_t^T)] \;\; subject\ to$$

$$\theta_{t+1}^S(s) = opt - alg_\theta(L^s(\theta_t^s, step^S)) \; and \; \theta_{t+1}^T(s) = opt - alg_\theta(L^T(\theta_t^T, step^S))$$

The optimization steps for θ^s and θ^t are set as 1 and based on paper observation θ^s can be approximated by θ^t. The equation can be simplified as

$$\min_{S} E_{\theta_0 \sim P_\theta}[\sum_{t=0}^{T-1} D(\nabla L^S(\theta_t), \nabla L^T(\theta_t))]$$

Instead of optimize the equation above with nested dependency, the method tries to generate the condensed set S such that the gradient of training loss on dataset S is similar to the ones on original set T at every iteration. In other word, the distance between 2 gradients are minimized. This optimization goal doesn't require unrolling of nested dependency graph to solve the optimization problem over previous parameters and its is more efficient in computation cost and memory usage.

2.1.5 1e: Usefulness of methodology in machine learning applications

The method of dataset condensation with gradient matching is considered as useful in machine learning applications because it is able to show effective learning of synthetic images can outperforms traditional coreset methods with a wide margin in multiple computer vision benchmarks. It also benefits other learning problems when there is a fixed budget on training images. Moreover, the method also outperforms popular data selection methods by providing more informative training samples in continual learning. Finally, there is a promising use case of the method in neural architecture search where once the condensed images are learned, they can be used to train numerous network architectures extremely efficiently.

2.2 Dataset Distillation Learning

2.2.1 2a) Benchmark Original Dataset

The lightweight ConvNet model is chosen as the default model to run benchmark test on both MNIST and CIFAR10 datasets. The model is trained for 20 epochs with batch size of 256. A cosine annealing scheduler is used with initial learning rate of 0.01. The FLOPs for the model is calculated with python package 'flopth'. For MNIST dataset, input size is given as 1 channel with height and width of 28 pixels. For CIFAR10 dataset, input size is given as 3 channels with height and width of 32 pixels. The model accuracy and FLOPs on the test sets are shown in table below. The training/testing FLOPs are calculated as the number of images in training/testing set multiplied by the FLOPs for a single image.

Table 1: ConvNet model test accuracy and FLOPs

Dataset	Accuracy(%)	Model FLOPs	Training FLOPs	Testing FLOPs
MNIST	99.06	48.90 M	2934.13 G	489.02 G
CIFAR10	72.34	51.26 M	2563.07 G	512.61 G

It can be observed that ConvNet can reach a good accuracy of 99.06 percent on MNIST dataset after 20 epochs. However, the CIFAR10 dataset is more complicated and ConvNet performs relatively poorer at 72.34 percent in generalizion and classification.

2.2.2 2b) Learn Synthetic Dataset and Gradient Matching Algorithm

The same 2 datasets are used to evaluate the performance of dataset distillation with gradient matching method using hyperparameter given in the project manual. The ConvNet model is used to perform the distillation task. The gradient matching method randomly picks 10 images per class from original the test set and use them as initial synthetic dataset S. Additionally, image augmentation is enabled with max crop factor of 4, scale factor of 0.2, rotation of 45 degrees and noise coefficient of 0.001. In every outer-loop step k, a new ConvNet is initialized with randomly distributed initial weights and bias. For every k, 10 inner loops T are performed. The loss gradients with respect to original dataset and synthetic dataset are computed individually and fed into cosine dissimilarity cost function for each image class. The synthetic data is optimized and updated for 1 step in GSD for each inner loop. Then, the ConvNet weights and bias parameters are trained for 50 epochs on the newly updated synthetic data.

2.2.3 2c) Visualization of Condensed Images

The distillation algorithm randomly selects 10 images per class from the training set as starting point for synthetic dataset. Over iterations, it optimizes the images based on the gradient loss dissimilarity function mentioned in section above. The visualizations are provided for both MNIST and CIFAR10 datasets below. The image sets at initial epoch and final epoch are listed side by side for direct comparison.

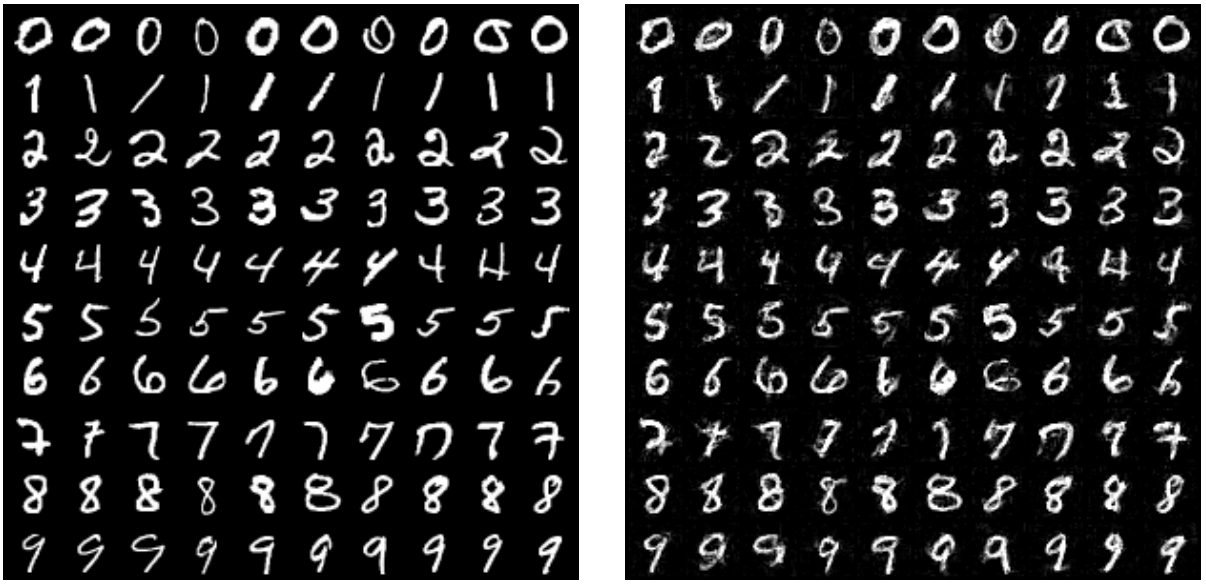

Figure 1: Visualization of condensed 10 image/class with ConvNet for MNIST (initialized from real image) a) initial epoch b) final epoch

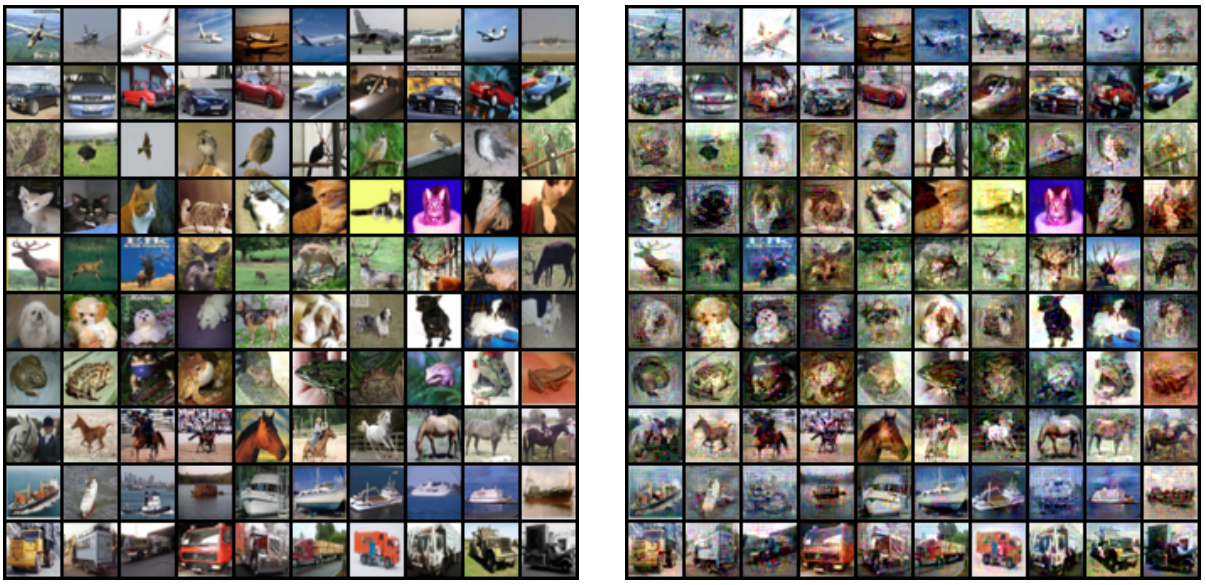

Figure 2: Visualization of condensed 10 image/class with ConvNet for CIFAR10 (initialized from real image) a) initial epoch b) final epoch

We can observe that the final condensed synethic sets for MNIST and CIFAR10 are still reminiscent to the initial ones with added noises and blurs. The image classes are easily recognizable. In Figure 1b, the images in class '7' are significantly modified and blurred from the initial ones. The samples at the center of the row looks more like '1' and suggests potential hidden patterns and similarity between class '1' and '7', which is not present in the initial set on the left. For the CIFAR10 dataset, visible RGB noises are prominent in the final synthetic dataset. These noises help the dataset to achive a more sophisticated representation of the object in different backgrounds and therefore provide better generalization information with limited number of images per class.

2.2.4 2d) Introduction of Gaussian Noise

The dataset distillation method is then tested on the random initialization. The synthetic dataset on the left are initialized with random Gaussian noise at first epoch. The hyperparameter settings are kept the same for comparison. The image results are shown below.

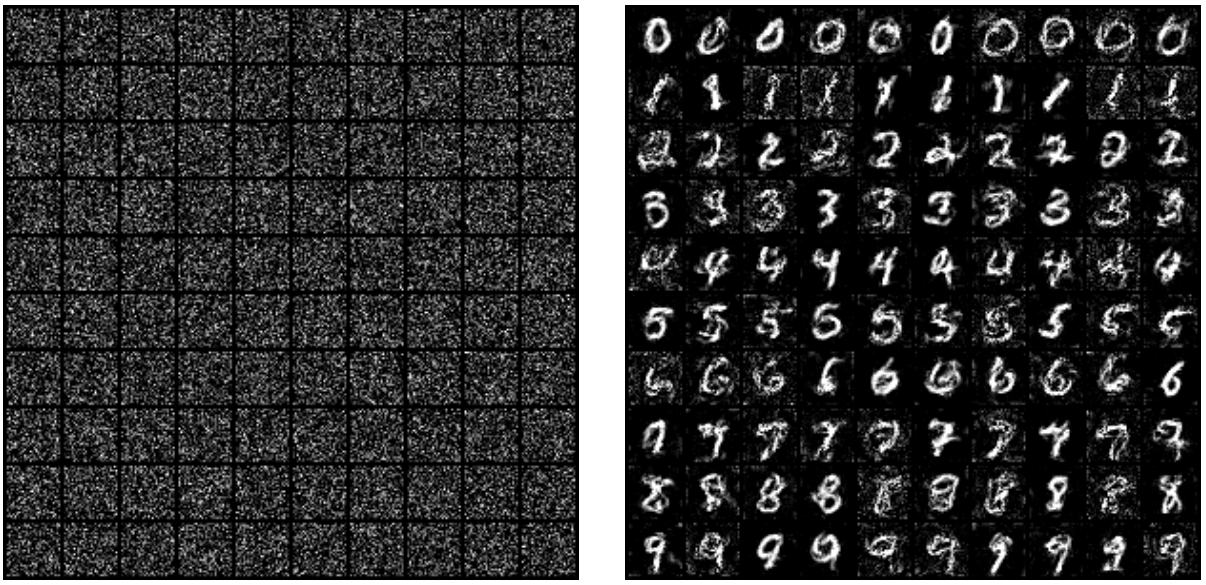

Figure 3: Visualization of condensed 10 image/class with ConvNet for MNIST (initialized from Gaussian Noise) a) initial epoch b) final epoch

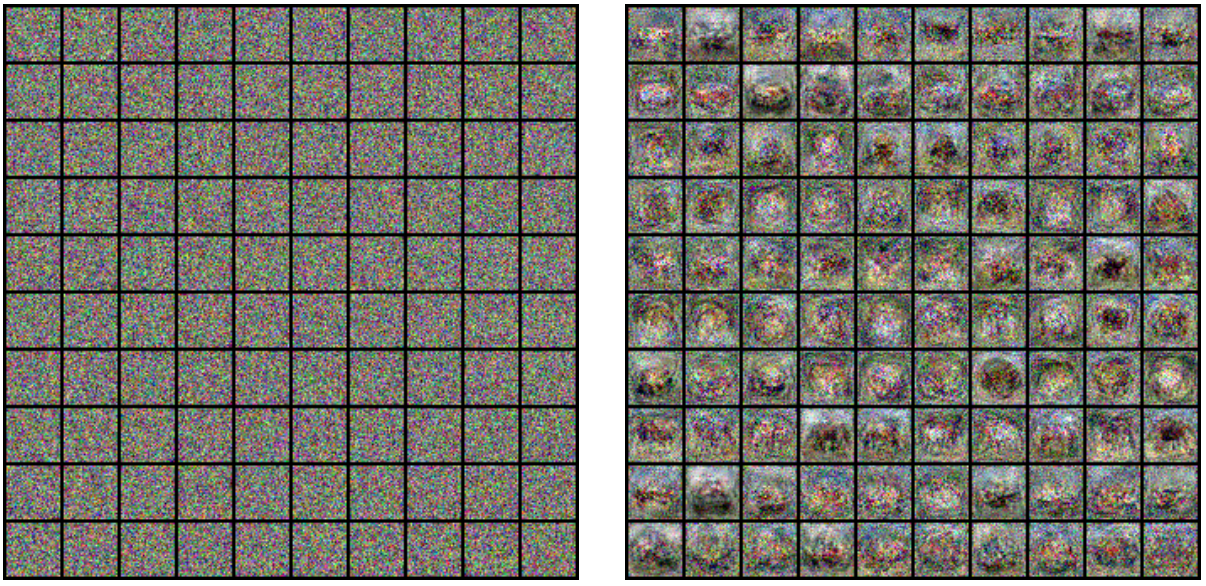

Figure 4: Visualization of condensed 10 image/class with ConvNet for CIFAR10 (initialized from Gaussian Noise) a) initial epoch b) final epoch

For MNIST dataset shown in figure 3b, the image classes are mostly recognizable and is significantly different from the ones in 3a. It can be seen that some of the numbers are overlaid with heavily salt and pepper noises and hard to recognize. In general, the images are embedded with some degrees of rotation and different forms of representation can be observed. Theoretically, it should still provide a good amount of generalization in class information. For CIFAR10 dataset shown in figure 4b, the classes of the images are not recognizable as it is overwhelmed by large amount of RGB noises in the background. Only the rough outlines of the image shape can be

observed.

2.2.5 2e) Train Selected Network from scratch on Condensed Images

The distilled datasets containing 100 training images in total each are evaluated against the original un-condensed MNIST and CIFAR10 datasets. The same ConvNet model is used for training from scratch on the 100 images distilled sets and all hyper-parameters are kept the same. The test set provided in original MNIST and CIFAR10 are used for accuracy evaluation. The results for condensed dataset initilaized from real images and random noises are shown along with the benchmark ones in the table below.

Table 2: Distilled dataset test accuracy against benchmark results

Dataset	Initialization	Accuracy(%)	Training time (secs)
Original MNIST	-	99.06	259.7
Distilled MNIST	Real images	94.39	5.3
Distilled MNIST	Random noises	94.07	5.2
Original CIFAR10	-	72.34	255.5
Distilled CIFAR10	Real images	38.50	21.7
Distilled CIFAR10	Random noises	35.37	21.6

The advantage of dataset condensation is that the model weight are decoupled from it previous states in training, hence requiring less training time once the images are trained. It can be seen in Table 2 that with the use of the synthesized data, the training time for both MNIST and CIFAR10 is reduced by 97.9% and 91.5% respectively, since learn learning rates are not learned for specific or each training steps. The gradient matching method of giving real and synthetic training images, helps avoid the expensive unrolling of the computational graph, significantly reducing the training time.

2.3 Cross Architecture Generalization

One of the key advantage of dataset distillation methods is that the condensed images learned using one architecture can be used to train another unseen one. In earlier sections of this project, the synthetic datasets for the MNIST and CIFAR10 have been learnt over the selected model, ConvNet. The synthesized set is then used to train another network listed in the networks.ipynb file to evaluate its cross-architecture performance in terms of classification accuracy on test sets. The new network selected to be trained is the LeNet.

Table 3: Cross-architecture performance in testing accuracy for condensed 10 ipc in CIFAR10

Distilled dataset	C/T	ConvNet	LeNet
MINIST from real images	ConvNet	94.39	88.27
MINIST from random noise	ConvNet	94.07	84.67
CIFAR10 from real images	ConvNet	38.50	25.83
CIFAR10 from random noise	ConvNet	35.37	24.24

Table 3 shows that the ConvNet condensed images for both the MNIST and the CIFAR10 dataset performed decently to an extend on the LeNet networks but not as well as the performance when it is trained with LeNet. However, the 88.27% and 84.67% performance on the MNIST dataset are good performances to be considered that the condensed images are architecture generic. It is harder for to determine for the CIFAR10 dataset as the original ConvNet trained performance was only 38.50% and 35.37% which is fairly low. It could be because that the LeNet architecture is not very suitable for the CIFAR10 dataset and more experimentation on other networks will have to be tested. For example, VGG11 is later tested and it obtained a performance of 32.21% which is fairly close to the performance on ConvNet, which is inline with its performance when trained on the original dataset.

2.4 Application. Apply your synthetic small datasets to one of the machine learning applications you proposed in part 1e

One of the applications for dataset distillation is neural architecture search [3] where the best model architecture (layer depth, width, etc.) is selected based on given evaluation metrics. The customized model should perform well without under-fitting or over-fitting. Traditionally, the search algorithm optimizes model architecture based on the training results and the optimization process can be extremely time consuming and computationally expensive. With distilled dataset, the computation cost to train a model is reduced exponentially. We can replace the dataset to optimize efficiency, if the metric of model trained on distilled dataset shows the same tendency as the original dataset. The search algorithm should yield the same results regardless of the datasets used. We conducted the experiment with 3 variations of the ConvNet model at different convolution layer depths and same width of 128 following model definitions in "utility.ipynb" file. The ConvNetD3 model is not investigate, as it is the same model used to distill MNIST dataset. We trained the model for 20 epochs using both datasets with cosine annealing scheduler and computed accuracy with original test datasets.

Table 4: Neural architecture search application

ConvNet variations	Original dataset accuracy(%)	Distilled dataset accuracy(%)
ConvNetD1	98.11	83.81
ConvNetD2	98.79	91.27
ConvNetD4	99.26	94.67

As shown in table 4, both datasets suggest the same trend in accuracy and indicate a posi-

tive relationship between ConvNet depth and test accuracy. The model ConvNetD4 yields the best results for both distilled and original dataset and is more suitable for applications related MNIST dataset. Although the test accuracy is different between the 2 datasets, the results provide the same knowledge in model picking (accuracy tendency) and suggest the same winner (ConvNetD4). Therefore, the distilled MNIST dataset is suitable for neural architecture searching applications and it can significantly reduce the searching cost without compromising in architecture selection results.

2.5 Dataset Distillation for Histophathological Classification Task

Table 5: Dataset Distillation with Gradient Matching for MHIST

Distilled dataset	C/T	ConvNet
MHIST from real images	ConvNet	0.0
MHIST from random noise	ConvNet	0.0

3 Task 2: Comparison with State-of-the-arts Methods - Trajectory matching

3.1 Task Overview

3.1.1 1a: What knowledge gap did your one/two chosen dataset distillation methods fill?

Dataset distillation with gradient matching [4] focus on short range matching on every step between the gradients computed from synthetic dataset and original dataset. The synthetic dataset is updated with only 1 training step at a time to match the original dataset. This method simplifies the nested optimization problem proposed in Dataset Distillation [1] and decompose the issue in to per step optimization. However, for evaluation of the dataset performance, a model is trained from scratch using the synthetic dataset over many steps and epochs. The error can accumulate and result in inferior test accuracy. The chosen paper "Knowledge distillation by matching training trajectory" instead focus on long range matching, which is closer to the evaluation process. The dataset is optimized based on the final parameter differences between expert and student over longer steps/epochs and mitigates the drawbacks from short range matching.

3.1.2 1b: What novelty did they contribute compared to their prior methods?

Instead of focusing on matching short range on every step, the paper focuses on long range by comparing the ending parameters between student (trained on synthetic dataset) and expert (trained on original dataset) parameters. The student network utilizes the same model as the expert and on every iteration, its parameters are initialized with selected expert parameters from a given starting epoch. This initialization ensures the student network to share the same starting point as the expert. Therefore, with an effective synthetic dataset, the student model should reach a similar ending set of parameters as the expert, but with fewer steps/epochs. This concept is converted into a lost function and used to optimize the synthetic dataset.

With sufficient iterations of this training loop and random initialization at different starting epoch parameters, the synthetic dataset can be optimized to support similar student training trajectories as the expert ones trained from original dataset. In addition, this method can distill all classes simultaneously and manage memory limitation more efficiently compared to the Gradient matching one class at a time distillation.

3.1.3 1c: Explain in full detail the methodologies of your selected methods.

Dataset distillation by matching training trajectory method involves 2 parts. First, a collections of expert trajectories (network parameters at each epoch) with different initializations are computed and saved. Then, during the dataset distillation process, the student network (same model selection as the expert) is initialized with the expert parameters (weights and bias) at a random starting epoch. The updated student network is trained for pre-determined number of steps on the synthetic/distilled dataset to further optimize the parameters. Lastly, the student network parameter are compared to the expert network parameters and the synthetic dataset is updated based on following loss.

$$L = \frac{||\hat{\theta}_{t+N} - \theta^*_{t+M}||_2^2}{||\theta^*_t - \theta^*_{t+M}||_2^2}$$

where t is the selected starting epoch, N is number of optimized steps for the student network and M is the number of expert epochs advanced from the starting epoch [4]. This loss penalizes on the differences between the updated student parameters $\hat{\theta}_{t+N}$ and the target teacher parameters θ^*_{t+M} with L2 norm and normalized by the teacher parameter progress. Noticeably the student optimized step is far less than number of expert epochs multiple by the number of steps per epoch. The synthetic dataset is learnt such that the student network trained on it can have similar parameters as the expert network trained on the original dataset, therefore ensures the usability and generalization capability of the distilled dataset.

3.1.4 1d: Discuss the main advantages and disadvantages of your selected methods. Do you think these methods can concretely distill the original datasets? Do you think your selected methods can analyze and inspect the cases of large-scale datasets like ImageNet [7]? Why?

Trajectory matching method produces better results due to long range focusing implementation and loss function. The dataset is optimized based on the final parameter differences between expert and student over longer steps/epochs and it can handle error accumulation over steps better. In addition, the method is more memory efficient, as it can distill all classes simultaneously in each training loop. However, this method relies completely on the expert trajectory, which needs to be computed on multiple experts over many epochs. This creation process requires extensive amount of resources and long training times based on number of experts selection. Also, the stored information can be demanding in disk space. This method is able to distill a synthetic dataset relatively well with majority of information and correlations encoded. However, it might not be suitable to distill large-scale datasets with the concern of expert trajectory creation. The creation process can scale exponentially on a large dataset and be extremely demanding in resources.

3.2 Similar to Task 1, the learning pipeline has two stages: (1) learn the condensed images using the selected method(s); (2) train your network from scratch on the condensed images, then evaluate them on the real testing data. (Make sure the comparison is fair.)

Dataset distillation by matching training trajectory method is investigated for this section. The more complicated dataset CIFAR10 is used for this experiment. To enforce a fair comparison with dataset distillation with matching gradient investigated in task 1, the same ConvNet model is used and the default data augmentation method is used. We selected 10 expert networks for trajectory creation and each expert is trained for 20 epochs to be resource efficient. After obtaining the expert trajectory, the dataset is distilled with 30 steps for student optimized steps N and 2 expert epochs M. The maximum starting epoch is set to 15 as the initial epochs contribute significantly to the optimization process. A total of 1000 iterations are computed, and distillation is performed with both initialization methods of from real images and from random noises. The final distilled dataset can be seen below.

Figure 5: Visualization of condensed 10 image/class with ConvNet for CIFAR10 (initialized from real image) a) initial epoch b) final epoch

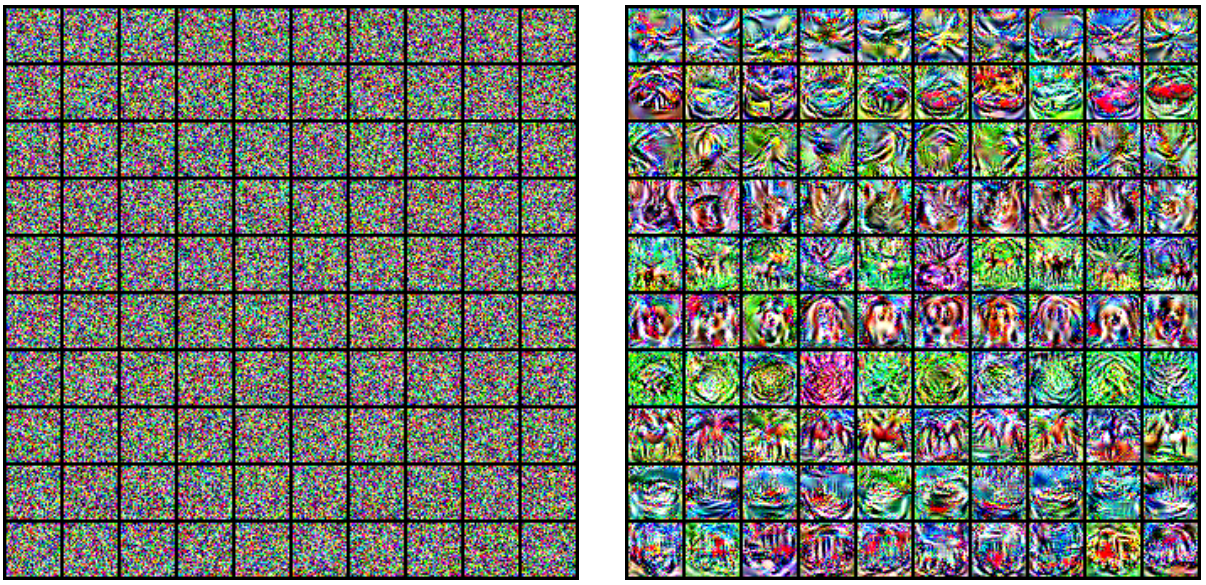

Figure 6: Visualization of condensed 10 image/class with ConvNet for CIFAR10 (initialized from Gaussian Noise) a) initial epoch b) final epoch

We can observe that in figure 5b, the final condensed image are still recognizable besides the initial one, with significant color distortion. In addition, some of the object shape has been altered and the image backgrounds are severely changed. Overall, the distilled image classes can still be inferred and the final condensed set is quite different from the one in the initial epoch in figure 5a. The distilled dataset initialized from random noise looks quite different from the RGB noise in the initial epoch. However, they maintain no resemblance to the image classes that they should represent. Only few contour lines can be seen from most of the images and the background is heavily contaminated by green and red color noises.

3.3 Report your findings in both quantitative and qualitative manners and compare them with the results of Task 1. Discuss your results. Do you think the selected methods outperform the Gradient Matching algorithm in terms of test accuracy? Explain the effect of the dataset distillation methods in terms of generalization and recognition abilities.

The distilled CIFAR10 datasets initialized from both real images and random noises are used to train a ConvNet model from strach for 20 epochs with the same setup as task 1. The original test dataset is used to evaluate the model accuracy.

Table 6: Distilled dataset test accuracy against benchmark results

Distillation Techniques	Initialization	Accuracy(%)
Undistilled/original	-	72.34
Gradient matching	Real images	39.19
Gradient matching	Random noises	36.39
Trajectory matching	Real images	45.75
Trajectory matching	Random noises	35.97

As can be seen from table, trajectory matching method outperforms gradient matching one by around 6.5% in accuracy for dataset initialized from real images. Trajectory matching performs long range matching at various starting point during distillation phase and it can handle error accumulation over steps better. This result is also in line with the visual observations from previous section. The distilled images have significant color distortion and and image backgrounds has been altered to provide more generalization information while maintaining the recognition ability for image classes. This allows the distilled dataset to embed more information with limited images per class. On the other hand, the distilled dataset from random noise using trajectory matching yields slightly worse results than the gradient matching one at 35.97% accuracy. We can see that random noises initialization performs worse for complicated datasets, as the dataset has no prior information at start compared to the other initialization method. Overall, trajectory matching performs optimization based on long range matching and yields better distillation results with proper initializations.

4 Task 2: Comparison with State-of-the-arts Methods - Distribution Matching Surrogate Objective

4.1 Task Overview

Dataset Condensation with Distribution Matching randomly chooses real and synthetic data, and then embed them with the randomly sampled deep neural networks. We learn the synthetic data by minimizing the distribution discrepancy between real and synthetic data in these sampled embedding spaces. [5]

4.1.1 1a: What knowledge gap did your one/two chosen dataset distillation methods fill?

Many dataset condensation methods still requires solving the expensive bi-level optimization which jeopardizes their goal of reducing training time due to the expensive image synthesis process. It also requires the tuning of multiple hyper-parameters such as the steps to update synthetic set and network parameters respectively in each iteration, that can be different for different settings such as sizes of synthetic sets.

Distribution matching data condensation method does not limit itself to its original data and can synthesize training images. Its synthetic data are optimized to match the original data distribution in a family of embedding spaces by using the maximum mean discrepancy(MMD) measurement. Distance between data distributions are commonly used as the criterion for

coreset selection, however, it has not been used to synthesize training data before. The method fills the knowledge gap of showing that the family of embedding spaces can efficiently be obtained by sampling randomly initialized deep neural networks.

4.1.2 1b: What novelty did they contribute compared to their prior methods?

The distribution matching method propose a novel training set synthesis technique that combines the advantages of previous coreset and dataset condensation methods while avoiding their limitations. Just like other dataset condensation methods, it does not limit itself to its original data and can synthesize training images. And like the coreset methods, it can efficiently produce a synthetic set and avoid expensive bi-level optimization. Its synthetic data are optimized to match the original data distribution commonly used as the criterion for coreset selection which has not been done before by prior methods. For example, the distribution matching condensation method is significantly faster than the gradient matching data condensation, and involves tuning only the learning rate for synthetic images of its hyper-parameter, while obtaining comparable or better results. Unlike other dataset condensation methods, the DM training is very efficient and can be independently run for each class in parallel and its computation load can be distributed. Due to its efficiency, it is also the first method to enable dataset condensation on larger settings and larger dataset.

4.1.3 1c: Explain in full detail the methodologies of your selected methods.

The goal of dataset condensation with distribution matching is to synthesize data that can accurately approximate the distribution of the real training data in a similar spirit to the coreset techniques. As the training images are typically very high dimensional, estimating the real data distribution can be expensive and inaccurate. In Distribution Matching, the training images are assumed to be embedded into a lower dimensional space by using a family of parametric functions ψ. Each of the embedding function ψ in the family is seen as providing a partial interpretation of its input, meaning their combination provides a complete one.

From this, the distance between the real and synthetic data distribution can be estimated using the empirical estimate of the maximum mean discrepancy (MMD) with the following equation:

$$\mathbb{E}_{\boldsymbol{\vartheta}\sim P_{\boldsymbol{\vartheta}}}\|\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\psi_{\boldsymbol{\vartheta}}(\boldsymbol{x}_i)-\frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|}\psi_{\boldsymbol{\vartheta}}(\boldsymbol{s}_j)\|^2$$

Next, the differentiable Siamese augmentation $\mathrm{A}(\cdot,\omega)$ are applied to real and synthetic data that implements the same randomly sampled augmentation to the real and synthetic minibatch in training, where $\omega\sim\Omega$ is the augmentation parameter such as the rotation degree. This enables the learned synthetic data to benefit from semantic-preserving transformations and learn prior knowledge about spatial configuration of samples while training deep neural networks with data augmentation.

The final step is to solve optimization problem of learning the synthetic data S by minimizing the discrepancy between two distributions in various embedding spaces by sampling ϑ.

As seen from the optimization equation, there are no model parameters involved and only S

$$\min_{\mathcal{S}} \mathbb{E}_{\substack{\vartheta \sim P_\vartheta \\ \omega \sim \Omega}} \| \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \psi_\vartheta(\mathcal{A}(\boldsymbol{x}_i, \omega)) - \frac{1}{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} \psi_\vartheta(\mathcal{A}(\boldsymbol{s}_j, \omega)) \|^2.$$

needs to be optimized, thus the equation can be effectively solved while avoiding expensive bi-level optimization.

The synthetic data are then trained for some iterations where in each iteration, the model will be sampled with a pair of real and synthetic batches and augmentation parameter for every class in the classfication. The mean discrepency between the augmented real and synthetic batches of every class is computed and summed as loss. The synthetic data is updated by minimizing the loss with stochastic gradient descent and learning rate.

4.1.4 1d: Discuss the main advantages and disadvantages of your selected methods. Do you think these methods can concretely distill the original datasets? Do you think your selected methods can analyze and inspect the cases of large-scale datasets like ImageNet [7]? Why?

The distribution matching method combines the advantages of previous coreset and dataset condensation methods by not limiting itself to individual samples from the original data and can synthesize training images while avoiding expensive bi-level optimization. The method is efficient as the synthetic data of different classes can be learned independently and in parallel. It is at an advantage compared to other methods as it is able to scale better with bigger and more challenging dataset because it does not have the complex nested loop optimization in terms of training time and memory usage. The method can concretely distill the original dataset since it is able to prove itself to produce more informative memory for continal learning and better proxy set for speeding up model evaluation in neural architecture search while achieving comparable or better performance. However, the method is at a disadvantage when compared with DSA's more data-efficient samples with a smaller number of synthetic samples. The reason for this is that the inner-loop model optimization in DSA with limited number of steps is more effective to fit the network parameters on smaller synthetic data. From the feedback given of the evaluation of the method on the TinyImageNet, even though the method has the efficiency to be applied to larger dataset, the method does not scale to the dataset well in terms of training time and memory usage due to the complex nested loop optimization. Therefore, it will pose a more challenging task to apply the method on an even larger dataset like the ImageNet. More research work will have to be extended for dataset condensation for more complex datasets and tasks in the future.

4.2 Similar to Task 1, the learning pipeline has two stages: (1) learn the condensed images using the selected method(s); (2) train your network from scratch on the condensed images, then evaluate them on the real testing data. (Make sure the comparison is fair.)

To ensure that the comparison is fair, all of the condensed images are learned with the same parameters and settings.

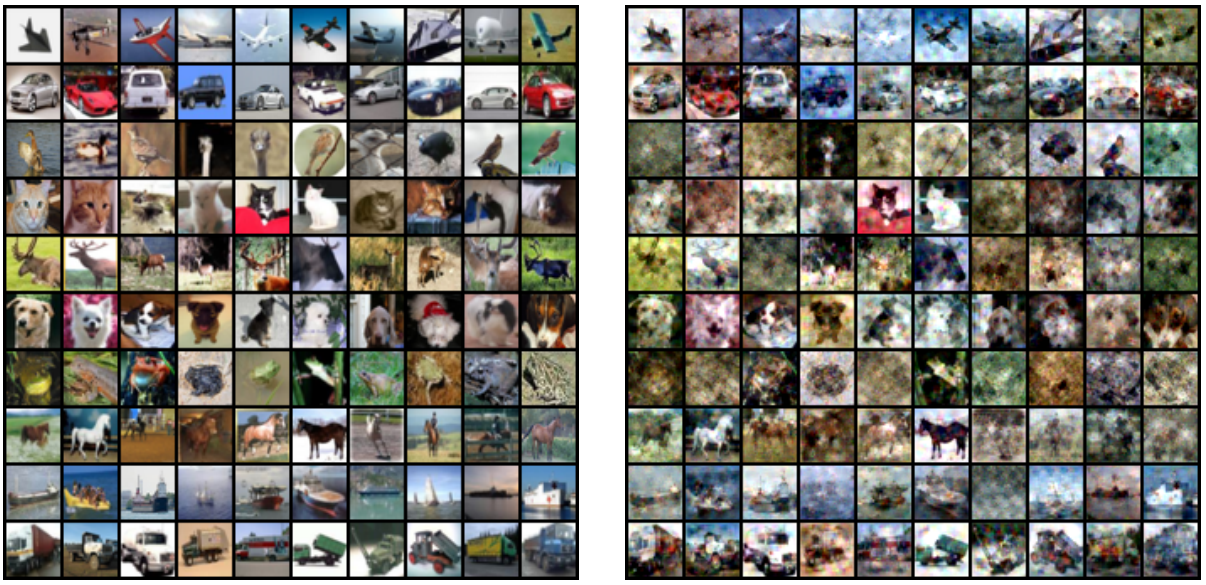

Figure 7: Visualization of condensed 10 image/class with ConvNet for CIFAR10 (initialized from real image) a) initial epoch b) final epoch

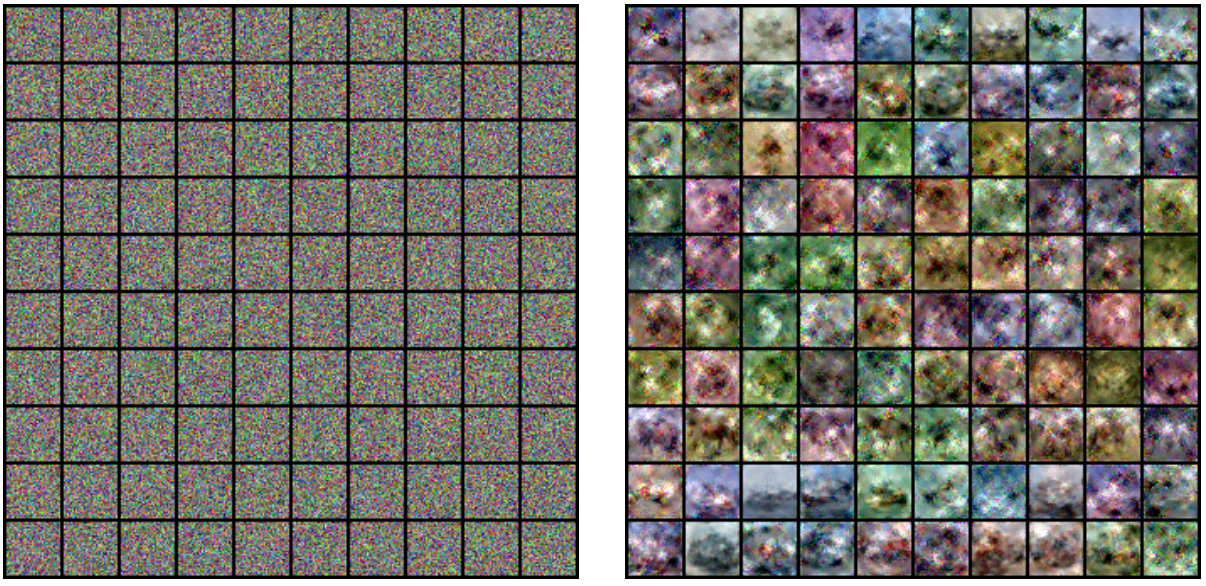

Figure 8: Visualization of condensed 10 image/class with ConvNet for CIFAR10 (initialized from Gaussian Noise) a) initial epoch b) final epoch

The learned synthetic images of CIFAR10 are visualized in Figure 7 and 8 initialized from both real original images and Gaussian noise. The synthetic images of CIFAR10 dataset from real images are visually recognizable and diverse. It is easy to distinguish the background and foreground object.

The synthetic images initialized from Gaussian noise also manage to generate some images that are not too obvious, but when inspected carefully, even though the image is more blurry and

contain some noise and unnatural strokes, some of the image can be recognizable as compared to some of the previous methods.

4.3 Report your findings in both quantitative and qualitative manners and compare them with the results of Task 1. Discuss your results. Do you think the selected methods outperform the Gradient Matching algorithm in terms of test accuracy? Explain the effect of the dataset distillation methods in terms of generalization and recognition abilities.

Table 7: Distilled dataset test accuracy against benchmark results on MNIST

Distillation Techniques	Initialization	Accuracy(%)
Undistilled/original	-	99.06
Gradient matching	Real images	94.39
Gradient matching	Random noises	94.07
Distribution matching	Real images	94.68
Distribution matching	Random noises	94.27

Table 8: Distilled dataset test accuracy against benchmark results on CIFAR10

Distillation Techniques	Initialization	Accuracy(%)
Undistilled/original	-	72.34
Gradient matching	Real images	39.19
Gradient matching	Random noises	36.39
Distribution matching	Real images	44.01
Distribution matching	Random noises	43.55

The method was evaluated on both MNIST and CIFAR10 datasets and their results are reported in Table 7 and 8 respectively. For both the MNIST and CIFAR10, they display similar pattern in their results and outperformed the Gradient Matching algorithm in terms of test accuracy. For MNIST, distribution matching achieved 94.68% accuracy with ConvNets which is better than the both gradient matching initialized with noise and original images. Similarly for CIFAR10, DM outperforms GM slightly by 6.62% but there still consists of a bigger gap between the method and the upper bound because the dataset contains more diverse images with varying foregrounds and backgrounds. Once again in DM, the performance of real images is still better than the performance of initialization from Gaussian noise but is significantly closer in value.

With the method of distribution matching dataset distillation when trained with different network parameter distributions, the networks that have lower validation accuracy will look blur and the images learned from networks with higher validation accuracy will look more colourful. But even though the synthetic images learned with different network parameter distributions look quite different, they have similar generalization performance. This is because these images are different in terms of their background pattern but similar in semantics. Hence the effect of the DM method is that it can can produce synthetic images with similar network optimization effects and generalization while significantly different visual effects. This explains

why that even though the final epoch in Figure 7 and 8 of the CIFAR10 dataset looks very different, they still managed to achieve an accuracy close to each other. Moreover, it is not the aim of this method for the synthesize dataset to be recognizable since regularizing the images to look real may limit the data-efficiency.

5 Conclusion

In conclusion, this project has achieved the objective of exploring the technologies and tools to create a synthetic small dataset that has the most discriminate features of the original large-scale dataset through various groups of dataset distillation methods. Specifically, the gradient matching, trajectory matching and the distribution matching objectives are served. In the first part of the project, the dataset distillation with Gradient Matching as a compression method have been successfully applied for MNIST and CIFAR10 and showed that the synthetic images are significantly more data-efficient than the same number of original images. They are also not architecture dependent and can be used to train different deep networks and to lower the memory print of datasets.

The distribution matching also proved as an efficient training set synthesis method which can produce more informative memory for continual learning and better proxy set for speeding up model evaluation in neural architecture search.

6 References

[1] Tongzhou Wang et al. Feb. 2020. URL: https://arxiv.org/pdf/1811.10959.pdf.

[2] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. "Dataset condensation with gradient matching". In: *arXiv preprint arXiv:2006.05929* (2021).

[3] Thomas G. *What is Neural Architecture Search? and why should you care?* Dec. 2021. URL: https://towardsdatascience.com/what-is-neural-architecture-search-and-why-should-you-care-1e22393de461.

[4] George Cazenavette et al. "Dataset distillation by matching training trajectories". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022). DOI: 10.1109/cvpr52688.2022.01045.

[5] Bo Zhao and Hakan Bilen. "Dataset Condensation with Distribution Matching". In: *arXiv preprint arXiv:2110.04181* (2020).

7 Appendix

Link to Github repository:

`https://github.com/hl5chen/ECE1512-2022F-ProjectRepo_YiyangShi_HsuanlingChen`

Python code and saved results can be seen in github repository under Project_B_Supp directory