ECE1512 Project A Report

Rémi Grzeczkowicz *MScAC Student*
University of Toronto
Student Number: 1010905399
remigrz@cs.toronto.edu

I. Introduction

II. Task 1

This part relies on the paper [3].

A. Part 1

(a) In this paper, the purpose of using Dataset distillation is to reduce the training cost while preserving the performance of the model.

(b) The advantages of their methodology over state-of-the-art are:
- It achieved unbiased representation of the real data distribution.
- It does not rely on rely on pre-trained network parameters or employ bi-level optimization.
- It has a reduced memory cost with a lower run time thanks to the fact that DataDAM does not use an inner-loop bi-level optimization.
- It outperformed other distillation methods except for one case where Matching Training Trajectory (MTT) performed better on CIFAR-10 with 10 Impage Per Class (IPC). MIT got an accuracy of $56.5\% \pm 0.7$ while DataDAM got $54.2\% \pm 0.8$.

(c) The novelty provided by this paper is the use of attention in data distillation. Indeed it has been used in knowledge distillation but never in dataset distillation.

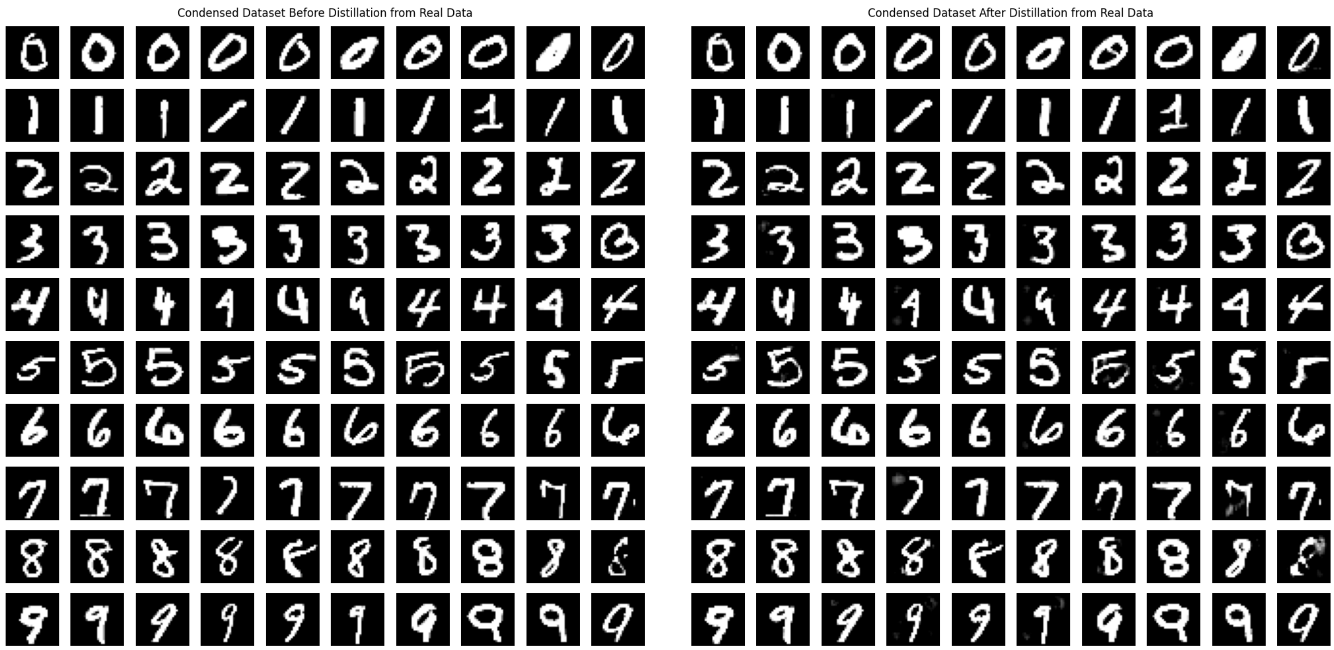
(d) The methodology is as follows:
1. (1) Initialize a synthetic dataset $\mathcal{S}$ either using random noise or sampling from the original training dataset $\mathcal{T}$.
2. (2) For each class k a batch B_T^k of real images and a batch B_S^k of synthetic images are sampled from $\mathcal{T}$ and $\mathcal{S}$ respectively.
3. (3) Then a neural network ϕ_θ is employed to extract features from the images. The network have different layers, each creating a feature map. This multiple feature maps allow to capture low-level, mid-level and high-level representations of the data.
4. (4) Using the feature maps of each layer, the Spatial Attention Matching (SAM) module generates an attention map for real and synthetic images. The attention map is formulated as $A(f_{\theta,l}^{T_k}) = \sum_{i=1}^{C_l} |(f_{\theta,l}^{T_k})_i|^p$ where $(f_{\theta,l}^{T_k})_i$ is the i-th feature map in the lth layer, C_l is the number of channels and p is a parameter to adjust the weights of the feature maps.
5. (5) The attention maps for both datasets are then compared using the loss function $\mathcal{L}_{SAM}$.
6. (6) The output of the network for each dataset is also compared using the loss function $\mathcal{L}_{MMD}$ based on the Maximum Mean Discrepancy (MMD).
7. (7) The total loss is then given by $\mathcal{L} = \mathcal{L}_{SAM} + \mathcal{L}_{MMD}$.
8. (8) Then $\mathcal{S}$ is updated such as $\mathcal{S} = arg\min_{\mathcal{S}} \mathcal{L}$.

(e) DataDAM could be used in machine learning for continual learning by providing an efficient memory management method by storing the synthetic data in the memory instead of the real data. This allows for a better memory usage and a lower computational cost. DataDAM could also be used for neural architecture search. Indeed, instead of training many architectures on the full dataset, those architectures could trained on the distilled dataset, leading to a faster search.

B. Part 2

1) Build the distillation model: To build the distillation model, we used the code provided by the author of [3] in the repo [1]. This allowed us to create a DataDAM class that we could use. Some change were required to make the understanding easier and to fit our framework as well as remove the uneeded parts.
In [1], we can notice that the authors introduced a factor 100 in their loss. After various attempt, and because they use more iterations than we do, we decided to set the factor to 10000 so that the loss is not too small and the dataset is updated correctly.

2) Distillation of MNIST dataset from real data: To generate the synthetic dataset associated with the MNIST dataset, we used the DataDAM class. The parameters used were : $model = ConvNet3D$, $IPC = 10$, $K = 100$, $T = 10$, $\eta_S = 0.1$, $\zeta_S = 1$, $\eta_\theta = 0.01$, $\zeta_\theta = 50$, $\lambda_{mmd} = 0.01$ and $minibatches_{size} = 256$.



(a) The synthetic dataset before distillation from real data

(b) The synthetic dataset after distillation from real data

Fig. 1: The synthetic dataset before and after distillation from real data

Figure 1a shows the synthetic dataset before distillation from real data and Figure 1b shows the synthetic dataset after distillation from real data. At first glance, we can see that the synthetic dataset is almost identical to the real dataset. To have a better understanding of the difference between the two datasets, we can look at Figure 2. This figure shows the difference between the synthetic dataset before and after distillation from real data. We can see that the difference then resides in the detail of the shape of the digits. This is a good sign as it means that the synthetic dataset condenses the information of the real dataset.

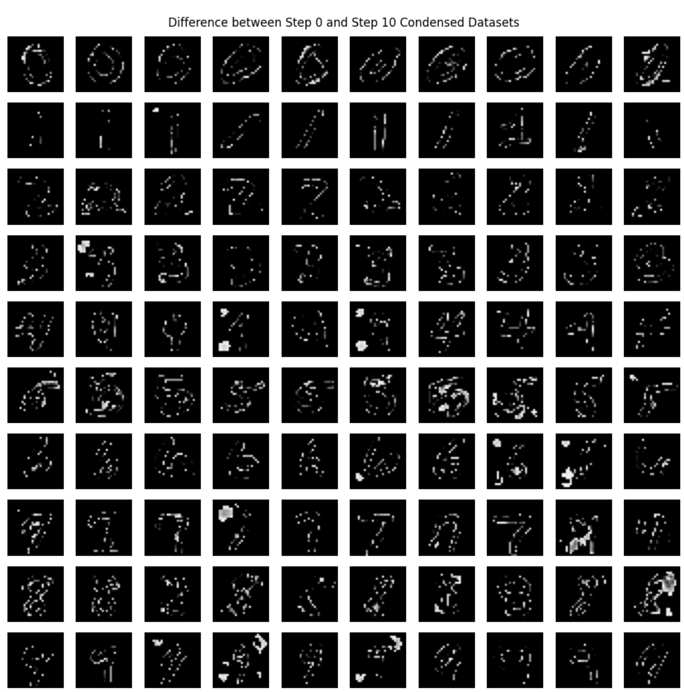
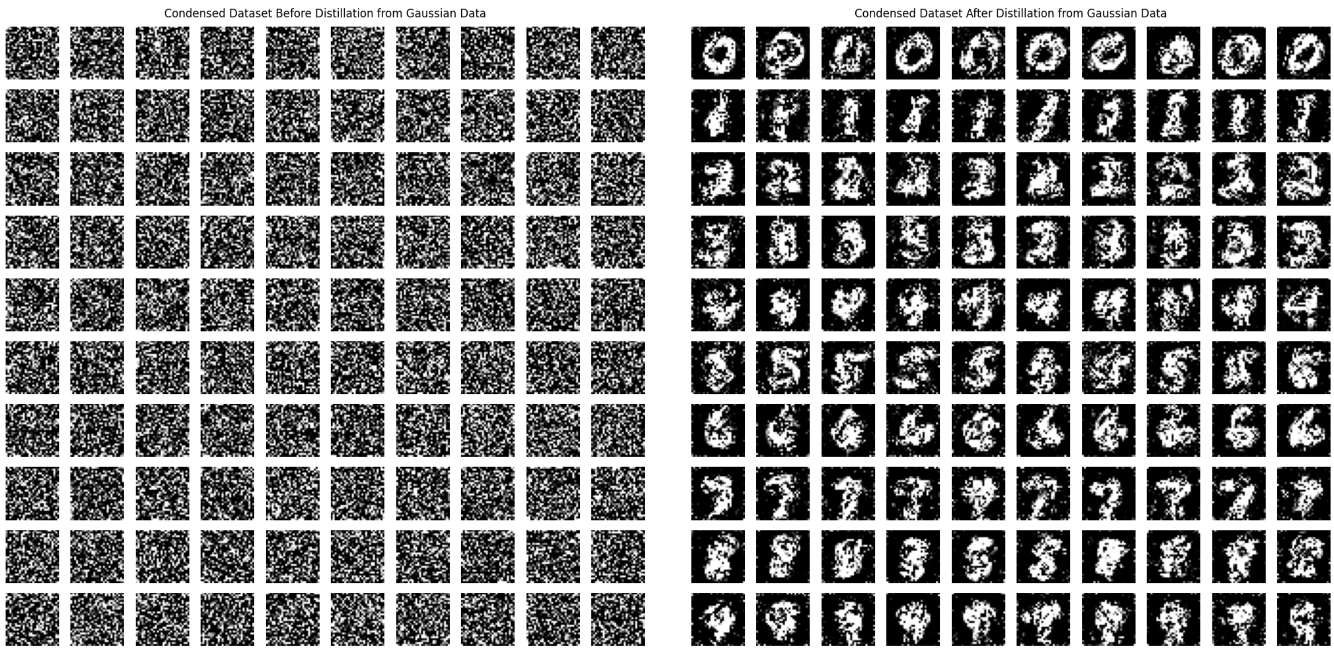


Fig. 2: The difference between the synthetic dataset before and after distillation from real data

3) Distillation of MNIST dataset from gaussian noise: To generate the gaussin noise, we used a standard normal distribution. The parameters used were the same as for the distillation from real data.



(a) The synthetic dataset before distillation from gaussian noise (b) The synthetic dataset after distillation from gaussian noise

Fig. 3: The synthetic dataset before and after distillation from gaussian noise

Figure 3a shows the synthetic dataset before distillation from gaussian noise and Figure 3b shows the synthetic dataset after distillation from gaussian noise. We can see that the distillation method succeeded in creating a synthetic dataset that shows digits we can recognize. However, those digits are not high quality.

4) Resutls and discussion for the MNIST dataset:

III. Task 2

This part relies on the paper [2].

A. Part 1

(a) The PAD method wants to solve the problem of misaligned information whether extracted or embedded into the synthetic dataset. The goal is to une only high-quality informations to improve the synthetic dataset.

(b) PAD introduced two novelties. The first one is the filtering of misaligned information during the extraction step by chosing the data to use according to the difficulty of samples and the image per class ration. The second novelty is the use of a layer-wise filtering in embedding. PAD only uses the deep layers of the agent model to perform distillation as they tend to captures higher quality information.

(c) The methodology is as follows:

(1) The real dataset is scored using a difficulty scoring function $\chi_t(x, y) = \mathbb{E}\left\|p(w_t, x) - y\right\|_2$ where x is a data point, y its label and $p(w_t, x) = \sigma(f(w_t, x))$ is the output of a model f transformed into a probability distribution.

(2) Using this difficulty score, a data scheduler is define as follow. All the easy data are selected first, then the hard data is gradually added to the set following the order of difficulty. When all the data is in the set, the easy data is gradually removed.

(3) An agent model is trained on the real dataset $\mathcal{D}_R$ using a Trajectory-Matching based method and the scheduler described. Then the change of parameters are stored. Let $\{\theta_t^*\}_0^N$ be a parameter sequence of the agent model (it is also called the trajectory).

(4) At each ineration, θ_t^* and θ_{t+M}^* are selected as start and target parameters. Let $\hat{\theta}_t$ be the parameters of the student agent model trained on the synthetic dataset $\mathcal{D}_S$. Then for N steps $\hat{\theta}_{t+i+1} = \hat{\theta}_{t+i} + \alpha\nabla\mathcal{L}_{CE}(\hat{\theta}_{t+i}, \mathcal{D}_S)$ where $\mathcal{L}_{CE}$ is the cross-entropy loss.

(5) To filter information $\hat{\theta}_{t+N}$ is defined such as only the $L-k$ last layers are used for matching where $k = \alpha * L$ and α is a hyperparameter that should be low in case of small IPC and high in case of high IPC.

(6) Then the synthetic dataset is updated such as it minimizes $\mathcal{L}$ where $\mathcal{L} = \frac{\left\|\hat{\theta}_{t+N} - \theta^*_{t+M}\right\|}{\left\|\theta^*_{t+M} - \theta^*_{t}\right\|}$.

(d) The advantages of PAD are: an improved alignment of information, a high-level of information (by discarding shallow layers) and, according to the paper, a better performance than other methods. The desadvanteges are the introduction of hyperparameter (notably α) that have to be tuned, the dependence of the difficulty scoring function that could be hard to define/improve. I believe that this method could distill the dataset correctly accordingt to the performances authors got. I believe that PAD will have more trouble with large scale dataset as the difficulty scoring function may score the entire dataset as difficult and thus leading to a non optimal scheduler.

REFERENCES

[1] GitHub - DataDistillation/DataDAM: [ICCV 2023] DataDAM: Efficient Dataset Distillation with Attention Matching — github.com. https://github.com/DataDistillation/DataDAM. [Accessed 21-10-2024].

[2] Zekai Li, Ziyao Guo, Wangbo Zhao, Tianle Zhang, Zhi-Qi Cheng, Samir Khaki, Kaipeng Zhang, Ahmad Sajed, Konstantinos N Plataniotis, Kai Wang, et al. Prioritize alignment in dataset distillation. *arXiv preprint arXiv:2408.03360*, 2024.

[3] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17097–17107, 2023.