ECE1512- Project A: Dataset Distillation: A Data-Efficient Learning Framework

Yingshun Liu 1006029049

Minghao Ma 1010800536

Content

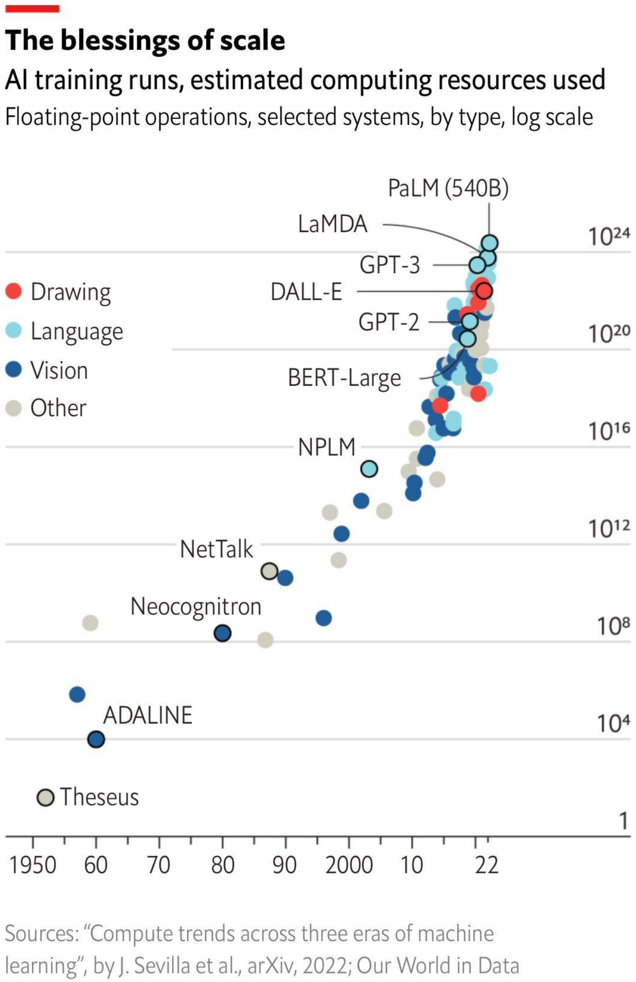
/01

Introduction

Background

- Deep learning has transformed many applications.
  - But training deep learning models on large datasets is compute-intensive.
  - It also requires a lot of storage and memory.

- **Dataset Distillation** reduces dataset size while retaining the essential information.
  - It reduces the storage and memory footprints but also speeds up training



Project Objectives

- Implement and evaluate a dataset distillation technique known as **Attention Matching**.

- Compared this technique against the state-of-the-art dataset distillation approaches, such as:
  - Prioritize Alignment
  - Distribution Matching

/02

Literature Review

Dataset Distillation Overview

- Dataset Distillation aims to reduce **dataset sizes** while retaining important information that allows models to perform well on downstream tasks.

- The main idea of this is the creation of a **synthetic dataset** that is far smaller compared to the original one, but achieves similar performance in training a model as using **the entire dataset**.

- Applications:

  - Privacy-Preserving Machine Learning
  - Resource-Constrained Environments
  - Efficient Data Transmission

  ...

Existing Dataset Distillation Methods

- Meta-Loss Based Dataset Distillation
- Gradient Matching Surrogate Objective
- Trajectory Matching Surrogate Objective
- Distribution/Feature Matching Surrogate Objective

/03

Task1: Attention Matching in Distillation

Dataset Distillation with Attention Matching

- Methodology and novelty of this paper:

  - Optimize the synthetic images to ensure that the attention distribution they generate aligns with that of the real data.
  - Adopts a dual-objective optimization to strike a balance between retaining data features and reducing data volume.
  - Ensures that its performance in the attention layer of the model is consistent with that of the real data through iterative optimization of the synthetic dataset

Experiments on MNIST and MHIST datasets

- Training ConvNet-3 on MNIST, ConvNet-7 on MHIST.
- The model was trained for 20 epochs using the SGD optimizer and a cosine annealing scheduler.

```
Epoch 1/20, Loss: 0.38628806681074995
Epoch 2/20, Loss: 0.10596167250992136
Epoch 3/20, Loss: 0.07172348202067487
Epoch 4/20, Loss: 0.059157884124904234
Epoch 5/20, Loss: 0.04938635293869896
Epoch 6/20, Loss: 0.04524999130913552
Epoch 7/20, Loss: 0.04037961067354425
Epoch 8/20, Loss: 0.036359604354947804
Epoch 9/20, Loss: 0.03221580803572656
Epoch 10/20, Loss: 0.029481215825542174
Epoch 11/20, Loss: 0.0274418665532102
Epoch 12/20, Loss: 0.024408989749412906
Epoch 13/20, Loss: 0.021954453821749764
Epoch 14/20, Loss: 0.02019625455339221
Epoch 15/20, Loss: 0.01841402786407382
Epoch 16/20, Loss: 0.016120270455374997
Epoch 17/20, Loss: 0.015017055652718594
Epoch 18/20, Loss: 0.013294779511287491
Epoch 19/20, Loss: 0.012989685234138147
Epoch 20/20, Loss: 0.011565992480857257
Test Accuracy: 98.39%
```

FLOPS:3976448

MNIST Performance

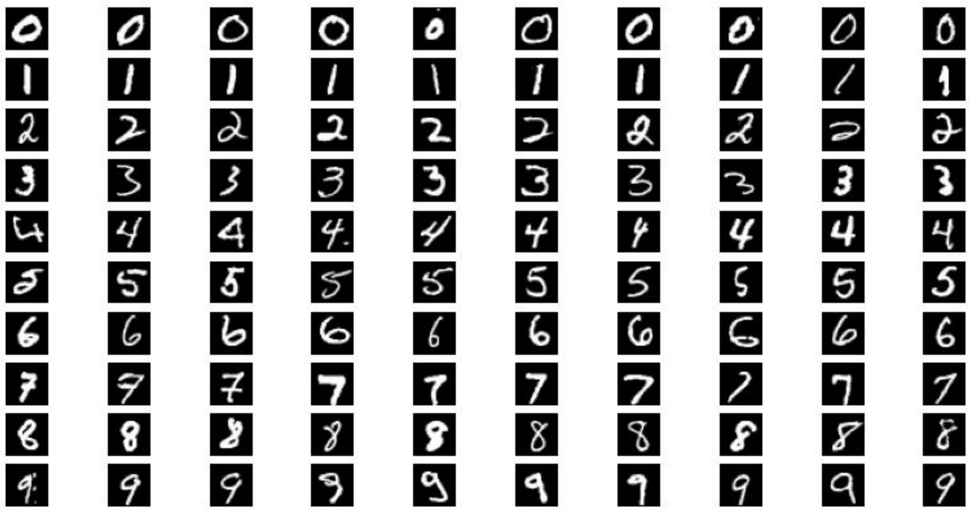
MHIST Performance

```
Epoch 1/20, Loss: 0.6353574776649475
Epoch 2/20, Loss: 0.6095033144950867
Epoch 3/20, Loss: 0.5861008203029633
Epoch 4/20, Loss: 0.5759060668945313
Epoch 5/20, Loss: 0.5436980748176574
Epoch 6/20, Loss: 0.5097814762592315
Epoch 7/20, Loss: 0.49354797720909116
Epoch 8/20, Loss: 0.4806488370895386
Epoch 9/20, Loss: 0.470728805065155
Epoch 10/20, Loss: 0.4637076282501221
Epoch 11/20, Loss: 0.4491611325740814
Epoch 12/20, Loss: 0.4270461857318878
Epoch 13/20, Loss: 0.4210186207294464
Epoch 14/20, Loss: 0.40984813213348387
Epoch 15/20, Loss: 0.39449386596679686
Epoch 16/20, Loss: 0.3647758162021637
Epoch 17/20, Loss: 0.355795601606369
Epoch 18/20, Loss: 0.3244953829050064
Epoch 19/20, Loss: 0.29357535362243653
Epoch 20/20, Loss: 0.2643807780742645
Test Accuracy: 92.95685279187818%
```

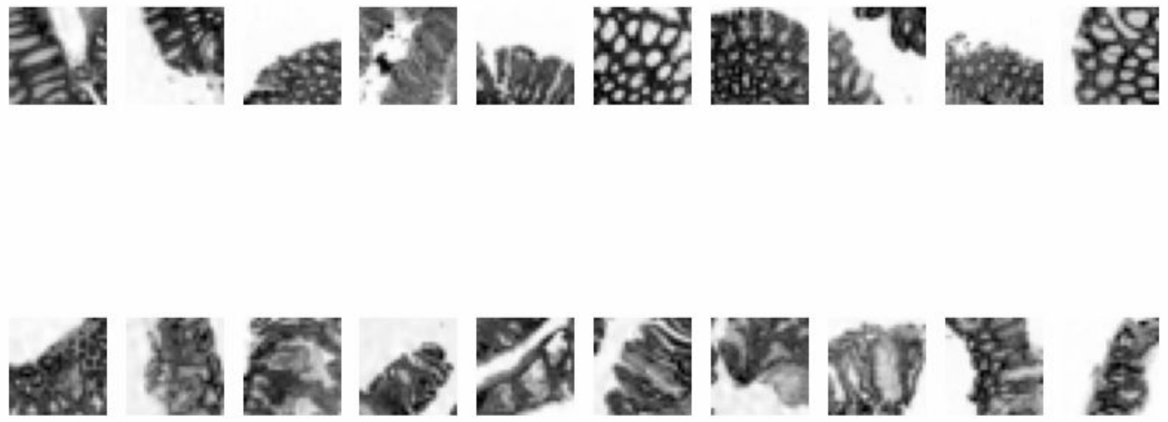
FLOPS:392226816

Visualization Results of the Synthetic Images

MNIST



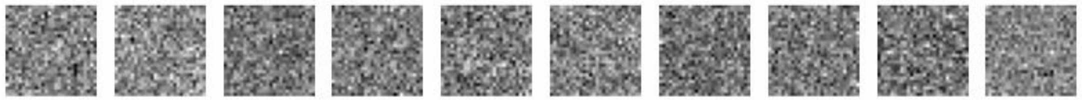
MHIST



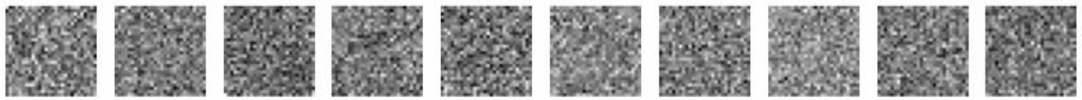
The condensed images were initialized by randomly selecting from real training images.

Repeat with Gaussian noise initialization

```
iter 1/10, Distillation loss: 0.693043692111969
iter 2/10, Distillation loss: 0.6930108022689819
iter 3/10, Distillation loss: 0.6929756879806519
iter 4/10, Distillation loss: 0.6929416799545288
iter 5/10, Distillation loss: 0.6929089403152466
iter 6/10, Distillation loss: 0.6928750658035279
iter 7/10, Distillation loss: 0.6928419804573059
iter 8/10, Distillation loss: 0.6928086471557617
iter 9/10, Distillation loss: 0.6927750110626221
iter 10/10, Distillation loss: 0.6927416086196899
Epoch 1/20, Loss: 0.6949266195297241
Epoch 2/20, Loss: 0.6777883768081665
Epoch 3/20, Loss: 0.6633500456809998
Epoch 4/20, Loss: 0.6496861577033997
Epoch 5/20, Loss: 0.635501503944397
Epoch 6/20, Loss: 0.620994508266449
Epoch 7/20, Loss: 0.6059054136276245
Epoch 8/20, Loss: 0.5906845331192017
Epoch 9/20, Loss: 0.5753730535507202
Epoch 10/20, Loss: 0.5600945949554443
Epoch 11/20, Loss: 0.5439456105232239
Epoch 12/20, Loss: 0.5273260474205017
Epoch 13/20, Loss: 0.51065593957901
Epoch 14/20, Loss: 0.4938836097717285
Epoch 15/20, Loss: 0.4766635000705719
Epoch 16/20, Loss: 0.4593597948551178
Epoch 17/20, Loss: 0.44201430678367615
Epoch 18/20, Loss: 0.42486149072647095
Epoch 19/20, Loss: 0.4077145457267761
Epoch 20/20, Loss: 0.3908585011959076
Accuracy on a real test set: 64.9746192893401%
```



Visualization results of the synthetic images for each class with Gaussian noise initialization



Gaussian noise initialization made it challenging for distilled images to capture real data features, resulting in poorer performance.

Cross-architecture Generalization

A different network architecture, LeNet, was trained on the synthetic dataset and evaluated on the test set.

The test results showed an accuracy of approximately 70%, indicating satisfied cross-architecture generalization of the synthetic dataset.

```
Epoch 1/20, Loss: 2.2556214332580566
Epoch 2/20, Loss: 2.2490291595458984
Epoch 3/20, Loss: 2.242401599884033
Epoch 4/20, Loss: 2.235717296600342
Epoch 5/20, Loss: 2.2290689945220947
Epoch 6/20, Loss: 2.2223877906799316
Epoch 7/20, Loss: 2.2157039642333984
Epoch 8/20, Loss: 2.209005832672119
Epoch 9/20, Loss: 2.202366590499878
Epoch 10/20, Loss: 2.195761203765869
Epoch 11/20, Loss: 2.189135789871216
Epoch 12/20, Loss: 2.1824707984924316
Epoch 13/20, Loss: 2.1758241653442383
Epoch 14/20, Loss: 2.169132709503174
Epoch 15/20, Loss: 2.1623597145080566
Epoch 16/20, Loss: 2.155492067337036
Epoch 17/20, Loss: 2.148563861846924
Epoch 18/20, Loss: 2.1415820121765137
Epoch 19/20, Loss: 2.134524345397949
Epoch 20/20, Loss: 2.1273608207702637
Accuracy on a real test set: 68.59137055837563%
```

/04

Task2: Prioritize Alignment in Distillation

Prioritize Alignment in Dataset Distillation

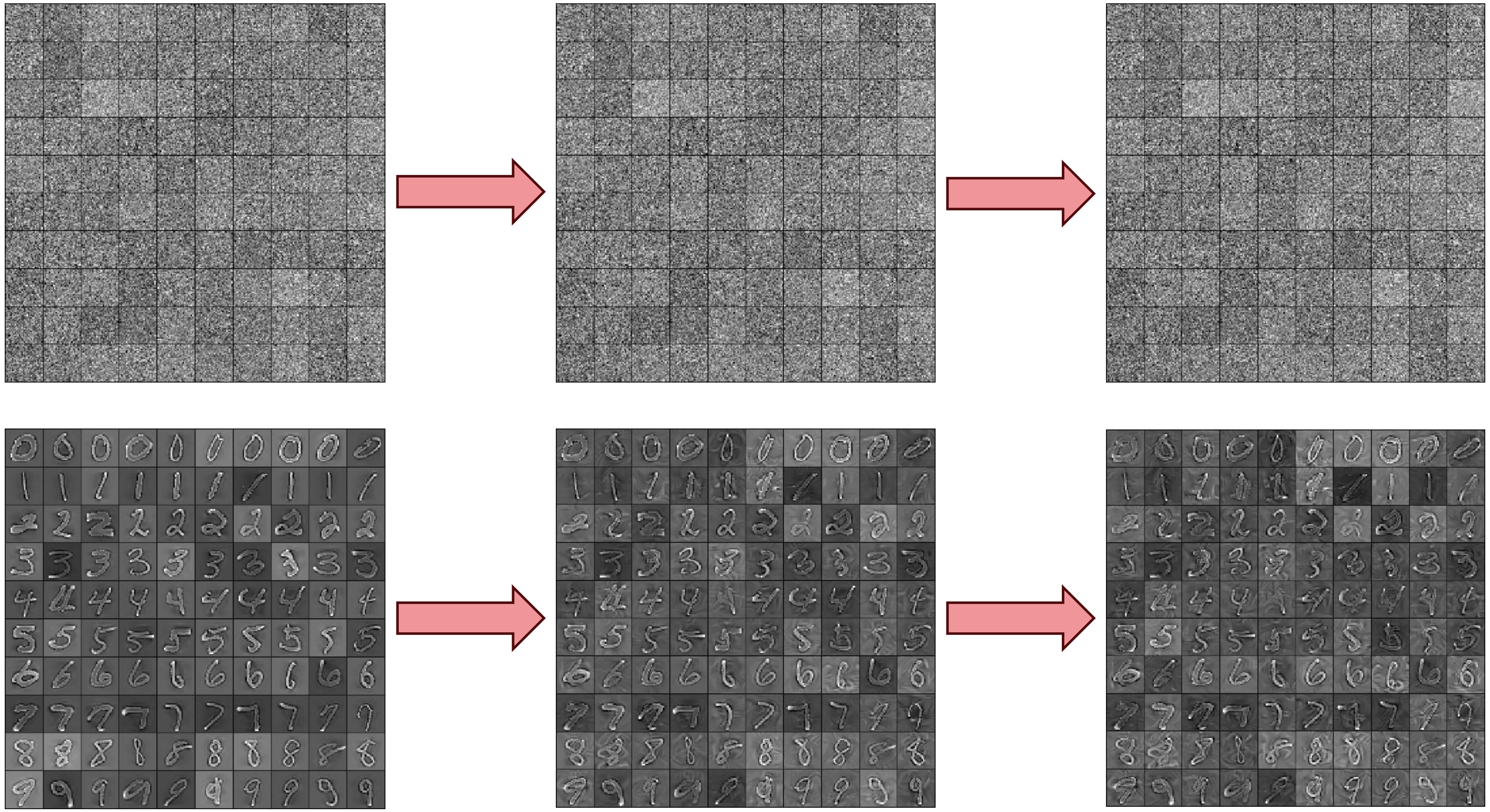
- Methodology and novelty of this paper:

  - Reveals the fact that both the information extraction and information embedding steps will introduce misaligned information.
  - Measures the difficulty of each sample in the target dataset, and employs a data scheduler to make sure the accessed data's difficulty is aligned with the compression ratio
  - Uses only parameters from deeper layers of the trained model to complete distillation.

Experiments on MNIST datasets

Initialize from random noise:

Initialize from correctly predicted samples:



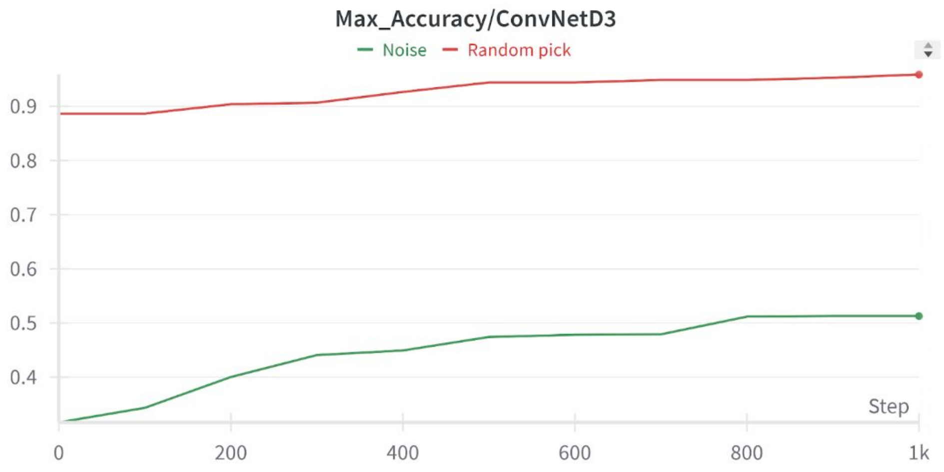
0 iteration

500 iterations

1000 iterations

Use ConvNet-3 as the agent model

Experiments on MNIST datasets



The max accuracy of model trained with synthetic images initialized by random noise: 0.5128

The max accuracy of model trained with synthetic images initialized by correctly predicted samples: 0.9588

Experiments on MNIST datasets

- The probable reasons why PAD didn’t perform well in this experiment:

  - The improper hyper-parameter selection

  - The convergence of the synthetic dataset

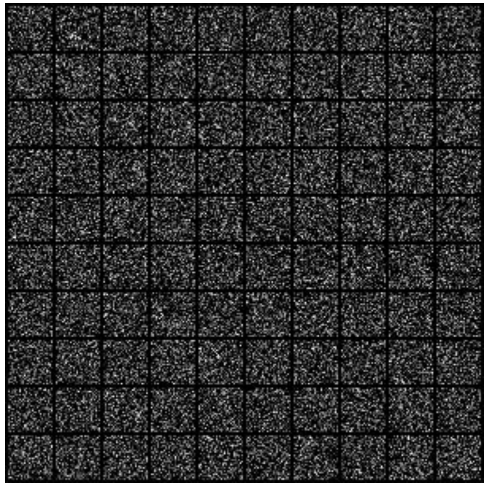
  - Other possible errors

/05

Task2: Dataset Condensation with Distribution Matching

Experiments on MNIST datasets

- The synthetic images were initialed by random noise.



0 iteration

Average test accuracy:
0.0885

500 iterations

Average test accuracy:
0.9716

1000 iterations

Average test accuracy:
0.9730

The decreasing loss trend indicates that the optimization effectively reduces the discrepancy between the synthetic and real data distributions, enhancing the quality of the synthetic dataset as training progresses.

Comparison between Task 1 and Task 2

- Test Accuracy:
  - Distribution Matching effectively aligns the generated synthetic data with the real data distribution, providing far superior test performance compared to Task 1.
- Convergence Speed and Loss Trend:
  - Distribution Matching's convergence is faster and more stable.
- Generalization Ability:
  - Distribution Matching achieves far superior generalization.
- Visual Quality of Synthetic Data:
  - Distribution Matching generates visually more coherent images.
- Training Time and Efficiency:
  - Distribution Matching requires more iterations to capture the data distribution.

Conclusion

Distribution Matching outperforms Attention Matching in test accuracy, convergence speed, generalization ability, and visual quality of the synthetic dataset.

Although Distribution Matching requires more training time, it produces synthetic data that closely approximates the real data distribution, making it the better option for high-accuracy tasks. Attention Matching, on the other hand, may be more suitable for scenarios requiring lower data quality and quicker results.

/06

Summary

Summary

- This work has investigated dataset distillation techniques, namely Attention Matching and Distribution Matching for creating compact synthetic datasets from larger ones while maintaining information that is essentially required for model training.
- These results indicate that Distribution Matching has outperformed Attention Matching on several fronts: test accuracy, convergence speed, generalization ability, and the visual quality of synthesized images.
- It seems apparent from practice that for Distribution Matching, high accuracy and generalization are achieved in tasks, while for situations where quick, low-cost solutions are feasible, Attention Matching could be of great help.

Thank you !