PRUNE THEN DISTILL: DATASET DISTILLATION WITH IMPORTANCE SAMPLING

Anirudh S Sundar [1] * *Gokce Keskin* [3] † *Chander Chandak* [2]
I-Fan Chen [2] *Pegah Ghahremani* [2] *Shalini Ghosh* [2]

[1]Georgia Institute of Technology [2]Amazon Alexa AI [3]Sanas AI, Palo Alto, CA, USA

ABSTRACT

The development of large datasets for various tasks has driven the success of deep learning models but at the cost of increased label noise, duplication, collection challenges, storage capabilities, and training requirements. In this work, we investigate whether all samples in large datasets contribute equally to better model accuracy. We study statistical and mathematical techniques to reduce redundancies in datasets by directly optimizing data samples for the generalization accuracy of deep learning models. Existing dataset optimization approaches include analytic methods that remove unimportant samples and synthetic methods that generate new datasets to maximize the generalization accuracy. We develop `Prune then distill`, a combination of analytic and synthetic dataset optimization algorithms, and demonstrate up to 15% relative improvement in generalization accuracy over either approach used independently on standard image and audio classification tasks. Additionally, we demonstrate up to 38% improvement in generalization accuracy of dataset pruning algorithms by maintaining class balance while pruning.

Index Terms— Dataset optimization, Dataset pruning, Dataset distillation, Deep learning

1. INTRODUCTION

Improvements in data collection pipelines have enabled the creation of large datasets for various tasks and have been a driving force for the success of deep learning models. However, operating on larger datasets costs additional storage and computational resources. Given these extrinsic constraints, it is important to understand whether all samples in large datasets contribute equally to improvement in model accuracy, especially when the increase in dataset size is associated with increased label noise and duplication. As a result, developing methods to reduce dataset size with minimal loss in generalization accuracy is an active research problem.

Based on existing research, dataset optimization methods can be categorized as either analytic or synthetic methods.

Analytic methods focus on selecting important data samples from a large dataset. Examples include removing samples that are forgotten during training [1], instance selection using herding [2], and dataset pruning by selecting data samples ranked using pruning metrics [3, 4].

In contrast, synthetic methods generate entirely new datasets optimized for a given task. Inspired by Knowledge Distillation [5] where information from a larger model is condensed into a smaller one, *Dataset Distillation* [6] is a technique to condense information from a large dataset into a smaller dataset of representative samples. Following the condensation process, a model trained on synthetic data was shown to generalize to unseen test data.

In this work, we first prune a dataset by carefully selecting a subset of samples representative of the original dataset, then further improve the information content of the pruned dataset with dataset distillation. We call this approach `Prune then distill`.

In Section 2, we discuss existing approaches to dataset optimization. In Section 3, we detail current state-of-the-art analytic and synthetic approaches and describe `Prune then distill`, our novel algorithm. In Section 4 we show that combining analytic and synthetic approaches yields better results than either one used independently. We also show that for high compression factors, pruning evenly across classes yields better results than uneven pruning, and that training on a specific percentile of EL2N scores yields the best results. Additionally, while prior work focuses on smaller 8-class datasets [7], we extend dataset distillation to 30 classes in the speech classification task.

2. RELATED WORK

2.1. Analytic methods

Examples of analytic methods include coreset methods, active learning, forgetting, and importance sampling. Coreset methods select data points to approximate a larger dataset in terms of performance on a downstream task, namely the generalization error of trained models [8, 9]. Active learning methods reduce the number of training points to label by intelligently selecting informative unlabeled samples [10, 11]. Forgetting methods compute the importance of a sample using the num-

*Main contact author: asundar34@gatech.edu. Work performed during an internship at Amazon Alexa

†Work performed while at Amazon Alexa

ber of times a label change occurs during training [1]. Importance sampling techniques rank data samples by computing and thresholding a scalar signal that quantifies importance, and prune datasets by retaining only the important samples [12, 13, 4, 3].

2.2. Synthetic Methods

Synthetic methods generate datasets optimized for a given task. *Dataset distillation* [6] creates a synthetic dataset of lower cardinality from a larger dataset. Optimizing synthetic data samples to reduce training loss over the original dataset ensures the satisfactory performance of synthetic data on downstream tasks. More recent work includes Label Distillation [14], Differentiable Siamese Augmentation [15], Distribution Matching [16], Gradient Matching [17], Synthetic-Data Parametrization [7], and Trajectory Matching [18].

3. METHOD

In this work, we show that combining analytic and synthetic dataset optimization methods achieve better results than either one used independently. In particular, we use dataset pruning followed by distillation using trajectory matching.

Dataset pruning methods select the most important samples to train a dataset by computing a scalar signal representing the importance of each data sample to model accuracy such as forgetting scores [1], norm of error vectors [4], proxy models [12], and memorization estimates [13].

Trajectory matching for dataset distillation [18], on the other hand, uses models trained on real data to distill additional knowledge into smaller, synthetic datasets. Minimizing the matching loss between trajectories of models trained on real and synthetic data is used to ensure that both models produce comparable results.

In Subsection 3.1 we discuss the EL2N score [4], a mechanism for pruning datasets. In Subsection 3.2 we discuss the trajectory matching algorithm for dataset distillation [18]. In Subsection 3.3 we detail `Prune then distill`.

3.1. Importance Sampling with Class Balance

The EL2N score [4] of a data sample is a scalar signal that quantifies how difficult it is to classify that data sample given a set of classifiers and determines its importance to model accuracy. Samples are ranked according to their EL2N score, and lowest scoring samples are removed from the dataset until a required compression factor is attained. Empirically, it is shown that removing these samples have the least impact on resulting classification accuracy [4].

The EL2N score of a data sample (x, y) using the weights of a model on the j-th iteration of training $\mathbf{w}_j$ is defined as the L2-norm of the difference between the predicted probability $p(\mathbf{w}_j, x)$ and the one-hot label y.

$$\textit{EL2N}(x, y) = \mathbb{E}_{\mathbf{w}} ||p(\mathbf{w}_j, x) - y||_2 \tag{1}$$

Consider a supervised classification problem consisting of a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=0}^{N}$, to be pruned. We train a set of classifiers $\mathcal{T}$ to classify the entire training dataset before pruning. We follow the same procedure as Paul et al. [4] with one modification; we prune the dataset evenly across classes. That is, we compute the EL2N score for all samples of each class. Then, based on the required compression factor, we select samples equally across classes to ensure class balance in the pruned dataset.

Algorithm 1 `Prune then distill`

Input: Dataset $\mathcal{D}$, Experts $\mathcal{T}$, K classes, N samples per class
Parameter: `compression` $\in [0, 1]$
Output: New dataset $\mathcal{D}_{new}$

```
% Dataset pruning
for t ← T do
    for (x_i, y_i) ~ D do
        Compute EL2N(x_i, y_i|t)
    end for
end for
EL2N(x_i, y_i) ← E_T[EL2N(x_i, y_i|t)] ∀i = [1, KN]
for k = 1 : K do
    Select {(x, y)} ~ D | y = k
    Sort {(x, y)} by EL2N(x_i, y_i)_{i=1}^N
    D_prune ← top compression factor of {(x, y)}
end for

% Dataset Distillation
D_new ← D_prune
for each distillation step do
    Sample expert t randomly from T
    Compute student trajectory θ on D_new
    Compute matching loss L(θ, t)
    Update D_new wrt L(θ, t)
end for
return D_new
```

3.2. Trajectory Matching

The trajectory matching algorithm [18] uses multiple pretrained expert trajectories to distill knowledge into a synthetic dataset. The synthetic dataset is first initialized with noise or random samples from the dataset. For each distillation step, an expert trajectory is selected from the list of pretrained trajectories. The student network is initialized with the weights of the expert at iteration j (θ_j^*) and is updated J times to reduce classification loss on the synthetic dataset. After the updates, a matching loss, given in Equation 2, is computed between the weights of the student after J iterations ($\hat{\theta}_{j+J}$)

and the weights of the expert after M iterations (θ^*_{j+M}) ($J \leq M$). Minimizing the matching loss encourages the weights of a model trained on synthetic data to be as close as possible to one trained on real data.

$$\mathcal{L} = \frac{||\hat{\theta}_{j+J} - \theta^*_{j+M}||}{||\theta^*_j - \theta^*_{j+M}||} \tag{2}$$

3.3. Prune then distill

As the name suggests, our method `Prune then distill` combines dataset pruning and distillation. First, we train expert models to classify training data. We compute EL2N scores for each training data sample given these expert models. We discard samples from each class with the lowest EL2N scores based on the required compression factor to obtain the pruned dataset while maintaining class balance.

Next, we use the pruned dataset as an initialization for the trajectory matching algorithm in place of random samples / Gaussian noise used by Cazenavette et al. [18]. We update the pruned dataset to minimize the matching loss described in Equation 2. Algorithm 1 details `Prune then distill`.

4. EXPERIMENTS

We study `Prune then distill` on CIFAR-10, CIFAR-100 [19], Tiny ImageNet [20], and Google Speech Commands [21]. Table 1 compares different approaches on the four datasets. We run experiments for 1000 distillation steps using 100 expert models trained separately for each dataset. `Prune then distill` obtains improvements on CIFAR-10, Speech Commands, and Tiny ImageNet for various compression factors. With larger synthetic datasets, the generalization accuracy of a model trained on this dataset increases since more information is available. Our approach does not yield noticeably better results on the CIFAR-100 dataset. We hypothesize that this is because 10% compression with 100 classes may be too aggressive for this dataset. While Tiny ImageNet has the same ratio of images per class as CIFAR-100, the images are of higher resolution, and a depth-4 ConvNet is used as opposed to a depth-3 ConvNet for the CIFAR experiments.

4.1. Impact of Initialization

The choice of initialization plays an important role in the dataset distillation algorithm. Initializing the synthetic dataset with random noise results in a 24% decrease in generalization accuracy when compared with initializing from random samples in the dataset, as seen from the results in Table 2.

4.2. Impact of Class Balance

While the original version of the dataset pruning algorithm removes samples based on their EL2N score, it disregards the fact that some classes may be more challenging than others to classify. We hypothesize that maintaining class balance while pruning datasets helps alleviate this issue. Table 3 compares the dataset pruning algorithm with and without class balance for 10% compression across datasets. Results from Table 3 indicate that pruning with class balance yields better results on CIFAR-100 and Tiny ImageNet, and near-identical performance on CIFAR-10.

At lower compression ratios, pruning based on only the EL2N score has the unintended consequence of modifying the underlying distribution of data samples across various classes, hurting the ability of models trained on pruned datasets to generalize to unseen test data. Table 4 presents the KL-divergence between the distribution of data samples across various classes before and after pruning. Table 4 shows that pruning based on EL2N score without class balance significantly changes the distribution of data samples across classes on CIFAR-100 and Tiny ImageNet (larger KL-divergences) and not as much for CIFAR-10 (smaller KL-divergence). In comparison to CIFAR-100 (100 classes) and Tiny ImageNet (200 classes), CIFAR-10 has far fewer classes (10 classes), resulting in more samples per class in the pruned dataset, improving the generalizability of trained models.

4.3. Cross-Architecture Portability

In line with previous dataset distillation algorithms, `Prune then distill` performs well on architectures other than the one used as expert models in the trajectory matching algorithm. Table 5 details the generalization accuracy of different model architectures trained on the pruned and distilled Speech Commands dataset. A ConvNet three times as large as the original expert architecture achieves better performance since there are more model parameters. The performance of an LSTM is comparable to the expert ConvNet with a similar number of parameters. The performance of an MLP is worse, which we attribute to the simplicity of the model's architecture. Still, the MLP performs better than random chance (3%), implying that the pruned and distilled dataset contains useful information from which other architectures can learn.

4.4. Impact of Sample EL2N Score

In addition to maintaining class balance during pruning, it is necessary to ensure that the pruned dataset is not corrupted by label noise. For aggressive compression factors (10%), many of the samples with the highest EL2N score may be ones belonging to incorrect classes. Utilizing these samples has the unintended consequence of hurting the performance of models trained on pruned datasets. To avoid this issue, we choose samples in the 70th - 80th percentile of EL2N scores from each class while pruning. Figure 1 presents the generalization accuracy of the same model trained on different subsets of CIFAR-100, where each subset is 10% of the size of the original dataset. Subsets are formed based on the EL2N score

Dataset	Compression (%)	Random Selection	Prune only	Distill Only	Prune-then-distill	Full Dataset
CIFAR10	0.02	14.4 ± 2.0	19.35 ± 0.3	44.5 ± 0.4	$\mathbf{44.7 \pm 1.5}$	84.8 ± 0.1
CIFAR10	0.2	26.0 ± 1.2	39.8 ± 0.4	62.7 ± 0.2	$\mathbf{63.1 \pm 0.7}$	
CIFAR10	1	43.4 ± 1.0	57.3 ± 0.3	68.5 ± 0.2	$\mathbf{69.7 \pm 0.4}$	
CIFAR10	10	69.6 ± 0.2	73.73 ± 0.2	69.8 ± 0.4	$\mathbf{76.11 \pm 0.3}$	
Speech Commands	10	60.8 ± 1.45	69.80 ± 0.9	67.96 ± 1.4	$\mathbf{71.23 \pm 0.9}$	85.3 ± 0.1
Speech Commands	20	75.43 ± 1.3	76.57 ± 0.7	76.33 ± 0.8	$\mathbf{77.21 \pm 0.5}$	
CIFAR100	0.2	4.2 ± 0.3	3.48 ± 0.2	$\mathbf{23.0 \pm 0.3}$	23.0 ± 0.2	56.2 ± 0.3
CIFAR100	2	14.6 ± 0.5	23.59 ± 0.2	35.8 ± 0.4	$\mathbf{36.2 \pm 0.5}$	
CIFAR100	10	30 ± 0.4	45.27 ± 0.2	45.4 ± 0.2	$\mathbf{45.5 \pm 0.5}$	
Tiny ImageNet	2	6.226 ± 0.2	12.4 ± 0.2	8.48 ± 0.775	$\mathbf{14.32 \pm 0.4}$	37.6 ± 0.4
Tiny ImageNet	10	17.0 ± 0.2	26.8 ± 0.3	23.56 ± 0.2	$\mathbf{27.66 \pm 0.3}$	

Table 1. Comparison of generalization accuracy of approaches on various datasets (in %)

Dataset	Noise Init	Sample Init
CIFAR-10	45.02	69.8
CIFAR-100	21.13	45.40
Speech Commands	49.12	67.96

Table 2. Random sample initialization performs better than random noise initialization for 10 % compression.

Dataset (# classes)	W/o Class Balance	Class Balance
CIFAR-10 (10)	**73.96**	73.73
CIFAR-100 (100)	36.2	**45.27**
Tiny ImageNet (200)	13.14	**26.8**

Table 3. Comparison of Dataset Pruning approaches, 10% compression. As the number of classes increase, pruning while maintaining class balance is better.

of the samples. The bin on the left represents 10% of data samples with the largest EL2N scores, while the bin on the right represents those with the smallest EL2N scores. From Figure 1, it is evident that training on 70th - 80th percentile of EL2N scores yields the best results. Training on samples with scores that are too high or too low is suboptimal. The highest scoring samples contain label noise, while the lowest scoring samples do not contain enough information for generalization.

5. CONCLUSION

In this work, we introduced `Prune then distill`, a dataset optimization method combining dataset pruning using EL2N scores and dataset distillation by trajectory matching. We showed that for a given compression factor, combining both approaches yields better generalization accuracy than either one used independently on CIFAR-10, Tiny-ImageNet, and Speech Commands. Additionally, we showed that pruning while maintaining class balance outperforms pruning without class imbalance for low compression factors.

Dataset	KL divergence
CIFAR-10	0.000856
CIFAR-100	0.008066
Tiny ImageNet	0.043840

Table 4. KL-divergence between distribution of data samples with and without class balance for 10% pruning.

Evaluation Model	Number of Parameters	Accuracy
Expert ConvNet	36k	71.23 ± 0.9
ConvNet	100k	77.72 ± 0.4
LSTM	34k	73.03 ± 0.2
MLP	33k	51.12 ± 0.06

Table 5. `Prune then distill` generalization accuracy across models on Speech Commands for 10% compression.

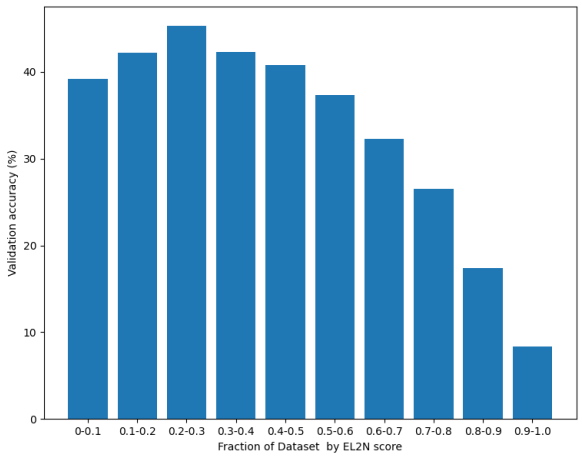

Fig. 1. Generalization accuracy using pruned CIFAR100 datasets selected using different fractions of EL2N scores

6. REFERENCES

[1] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon, "An Empirical Study of Example Forgetting during Deep Neural Network Learning," Nov. 2019.

[2] Yutian Chen, Max Welling, and Alexander J. Smola, "Super-Samples from Kernel Herding," *CoRR*, vol. abs/1203.3472, 2012, arXiv: 1203.3472.

[3] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos, "Beyond neural scaling laws: beating power law scaling via data pruning," June 2022.

[4] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite, "Deep Learning on a Data Diet: Finding Important Examples Early in Training," July 2021.

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the Knowledge in a Neural Network," Mar. 2015.

[6] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros, "Dataset Distillation," Feb. 2020.

[7] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song, "Dataset Condensation via Efficient Synthetic-Data Parameterization," June 2022.

[8] Stanley J Reeves, Larry P Heck, and Menlo Park, "Selection of Observations in Signal Reconstruction," *IEEE Transactions on Signal Processing*, vol. 43, no. 3, pp. 788–791, Mar. 1995.

[9] Dan Feldman, "Introduction to Core-sets: an Updated Survey," Nov. 2020.

[10] Ozan Sener and Silvio Savarese, "Active Learning for Convolutional Neural Networks: A Core-Set Approach," 2017.

[11] G. Tur, R.E. Schapire, and D. Hakkani-Tur, "Active learning for spoken language understanding," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, Apr. 2003, vol. 1, pp. I–I, ISSN: 1520-6149.

[12] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia, "Selection via Proxy: Efficient Data Selection for Deep Learning," Oct. 2020.

[13] Vitaly Feldman and Chiyuan Zhang, "What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation," Aug. 2020.

[14] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales, "Flexible Dataset Distillation: Learn Labels Instead of Images," Dec. 2020.

[15] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen, "Dataset Condensation with Gradient Matching," *arXiv:2006.05929 [cs]*, Mar. 2021.

[16] Bo Zhao and Hakan Bilen, "Dataset Condensation with Differentiable Siamese Augmentation," June 2021.

[17] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen, "Dataset Condensation with Gradient Matching," *arXiv:2006.05929 [cs]*, Mar. 2021, arXiv: 2006.05929.

[18] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu, "Dataset Distillation by Matching Training Trajectories," *arXiv:2203.11932 [cs]*, Mar. 2022.

[19] Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," p. 60, 2009.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[21] Pete Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," Apr. 2018, arXiv:1804.03209 [cs].