This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/PAINE56030.2022.10014786, 2022 IEEE Physical Assurance and Inspection of Electronics (PAINE)

Dimensionality Reduction to Dynamically Reduce Data

Dominic Sanderson
Department of Electronic and Computer Engineering
Brunel University London
London, England
dominic.sanderson@brunel.ac.uk

Ben Malin
Department of Electronic and Computer Engineering
Brunel University London
London, England
ben.malin@brunel.ac.uk

Tatiana Kalganova
Department of Electronic and Computer Engineering
Brunel University London
London, England
0000-0003-4859-7152

Richard Ott
Air Force Research Laboratory
Sensors Directorate
Ohio, United States

***Abstract*—We perform experiments using dynamic data reduction on datasets of moderate complexity, with focus on classification of a Micro-PCB image dataset. As deep learning models increase in complexity, the data that they use increases at a rate we can't keep up with. The result of this is often slight improvements to the model's accuracy, at the improportional cost of computational runtime, which increases the electricity used, and ultimately carbon emissions. By using data reduction techniques, we attempt to identify the least critical data to be excluded from training, which in turn cuts the environmental cost. We show the effect of data reduction techniques on moderately complex image data, including PCB images, to reduce runtime by 2% and improve the accuracy by 0.013%.**

***Index Terms*—Data reduction, Dynamic Data Reduction, UMAP**

I. Introduction

With every success of machine learning, the ambition and complexity increases as we dive deeper into its possibilities. However, as we increase complexity, we increase the need for more data to fuel our network models. Such practice is steadily becoming unsustainable, not only for data collection, but for network training [1].

Current trends within the AI industry are shifting from model-centric AI towards data-centric. The focus is changing from how we can improve the model for performance to how we can optimise our data for it. As we focus more on what data is best for our models, we turn our focus to what data we need most, and what data we do not [2] [3].

In this paper, we analyse both static and dynamic data reduction, using non-random data exclusion. We compare these methods across two datasets of moderate training difficulty, for the task of image classification.

The paper is structured as follows: Section II discusses related work, and the methods and experiments are described in Sections III and IV respectively. Section V discusses the results, and Section VI concludes the study.

II. Related Work

A. Dimensionality Reduction

For the task of image classification, the primary goal is to differentiate images between classes. One might attempt to plot each data point in a dataset, so that the borders that separates the classes might be found. However, the number of dimensions that an image represents cause near uniformity of distance from the classes' centroid. This problem is known as the curse of dimensionality [4]. Consider the handwritten image dataset, MNIST; each image is 28 pixels square with a single color channel, meaning the dataset is 784-dimensional. This becomes even more of a challenge as the size of data increases, where the number of dimensions ranges in the several hundred thousand.

To counter the effects of high dataset dimensionality, more data points may be useed for neural network training. However, this is difficult for some datasets, where data is either sparse, suffers from class imbalance, or lacks variation. For example, medical image datasets are both sparse and suffer from class imbalance [5], and PCB datasets have few images per class with little difference between them, due to their design.

Alternatively, Uniform Manifold Approximation and Projection (UMAP) is a dimensionality reduction algorithm that has been used widely for pre-processing in machine learning tasks [6]. It is applicable to many types of data and has been greatly successful in tasks including image classification [7], medical science [8], and electronics [9] [10]. By using UMAP to reduce the dimensionality of a dataset, the distance between each data point may be much more easily quantified, which allows further processing tasks such as data reduction.

B. Static Data Reduction

Data reduction is uncommon practice in image classification tasks, as more often, more data leads to better classification. However, in cases where some data is undesired, it is beneficial to remove it; for example, data trimming is the process of removing extreme values, without comprimising the dataset [11].

Research in data reduction has shown that data may be categorised using dimensionality reduction. By displaying data points in Euclidian space, the data points may be viewed as 'near to' or 'far from' the classes' centroid. With this information, data may be excluded based on its distance from its classes' centroid. This method is effectively data trimming, where the valid data range is determined by distance from each data points' classes' centroid.

Instead of reducing the network's performance, as expected due to fewer data points being used for training, the accuracy of the classification may be improved [12]. On top of bonus accuracy, by training with less data we reduce the runtime of the training, which correlates to less spent on electricity, which reduces the carbon footprint of the whole training process.

C. Dynamic Data Reduction

There has been recent research into dynamic data reduction [13]. Traditionally, the same quantity of data is used throughout training. However, by using less data, both the training time and computational resources may be reduced, at the cost of classification accuracy. Also, by excluding some data, the model may lose features vital for classification. To counteract this, it is possible to select a fraction of the data to use for a fraction of the training, and increase or decrease this amount of training data as training progresses. This ensures all data points of a dataset are utilised, but some data points of less importance are used less for training. The question is, therefore, how we determine the importance of a single data point relative to its peers.

The experiments described in this paper investigate data reduction by means of central and lateral data exclusion. We also investigate the use of these methods combined with simple dynamic data reduction methods based on work by [13]. Previous works focus on adaptive pruning, whereby data is removed at intervals during training, depending on its performance [14]. This methodology differs from this by defining the data reduction process prior to the start of training. The reduced dataset is used for the first half of training, and then the full unpruned dataset is used for the remainder of training. The contributions of this paper are as follows:

- A novel combination of dynamic data reduction and lateral/central exclusion is explored
- Testing on datasets of moderate complexity is performed and analysed
- Evidence is given that the combined techniques used are an improvement on standard data reduction.

III. Methods

A. Datasets

Two datasets of moderate training difficulty were used for training. They were chosen in order to evaluate the effectiveness of data reduction techniques on more complex data than those used previously [13], as well as to experiment on PCB image classification specifically.

The first dataset is the Micro-PCB dataset, which consists of 6500 training images, 1625 test images. Images are downscaled to 299x299x3 for training and testing. The data is split across 13 classes with even class distribution. Each PCB has been photographed from 25 different perspectives, and 5 unique rotations. This gives images in each class much more variation, as images of PCBs from a single perspective would offer very little variation.

The second dataset is Imagenette, which consists 9,469 training images, and 3,925 test images. Images were scaled to 299x299x3 for training and testing. The dataset is split into 10 classes with between 858 and 993 images per class.

B. Hardware

All experimentation was performed on the same hardware, for maximum consistency. The CO_2 usage was computed with an online calculator tool [15].

- CPU: Intel i7 10700
- GPU: RTX 4000
- RAM: 64GB
- CO_2 emission per hour: 0.0922kg

C. Performance Metrics

The experiments were performed to observe the effect of reducing data used each epoch of training. To quantify this, the experimentation results provide the total number of evaluations performed, where an evaluation is a single image used for training. The runtime of each experiment is also recorded, to show how the data reduction method used affects not only the accuracy of the network, but the runtime. The runtime and accuracy for each experiment is directly compared with the baseline for each dataset, to show the increase or decrease in performance in accuracy and runtime.

The p-values of each experiment are also calculated to show the statistical significance of each resulting accuracy. The closer this value is to 0, the greater the statistical significance. The p-values were calculated based on analysis of variance (ANOVA), using the accuracy results of the baseline experiment (where the full dataset is used, with no reduction) and each experiment with data reduction, individually.

As the model is otherwise unchanged when run with data reduction techniques, the CO_2 reduction of each experiment is directly correlated to the runtime. Therefore, CO_2 reduction is equal to the runtime reduction.

D. Network Architecture

The model used for testing is a Convolutional Neural Network with a Simple Monolithic Architecture. It uses 11 Convolutional Layers, and a primary and secondary Capsule layer. The Capsule layer uses Homogeneous Vector Capsules, which replace the fully connected layer, based on the model used by [12]. This model shows relatively high performance despite its few network layers.

Tests were run for 250 epochs with a batch size of 120. Optimisation was performed using the Adam optimiser with an initial learning rate of 0.98, and an exponential decay rate of 0.001 per epoch. These values provide consistent initial settings across all experiments for all datasets.

Online data augmentation was used during the training. The augmentation used was uniform, and the same methods were used as those performed in experiments in [12].

Citation information: DOI10.1109/PAINE56030.2022.10014786, 2022 IEEE Physical Assurance and Inspection of Electronics (PAINE)

E. Data Selection

By using UMAP for dimensionality reduction, the data points are represented in Euclidian space, making each data point observable in relation to every other data point in the dataset. This in turn allows us to include or exclude data points depending on their Euclidian distance to the classes' centroid. We use this method to perform experiments to show the effect of exclusion of data closest to the classes' centroid, and the data furthest. Previous work has shown that some datasets perform better with data close to the centroid, and others with data further, which we investigate for the two datasets chosen. Experiments performed exclude 1%, 5% and 10% of the data.

We define Static Data Exclusion as data exclusion that is the same throughout training. Before training is started, the data to be used for training is selected through the data selection process, and throughout each loop of training the same reduced dataset is used.

We define Dynamic Data Exclusion as data exclusion that changes in some way during training. For the experiments performed in this paper, we employ a 'Data Step' [13] method to increase the amount of data used during training, at a given point in training. Prior to that point, a reduced dataset is used to train with.

IV. Experimentation

A. Benchmark

Table I shows the benchmark runtimes and accuracies for both datasets used. These values will be compared against the results of all experiments to give a reference to the change in runtime and accuracy of each data reduction technique.

TABLE I: Benchmark accuracies and runtime

Dataset	Evaluations	Runtime (hours)	Test Accuracy	
			Average	Standard Deviation
Micro-PCB	1,620,000	1:55	99.96%	5.73E-04
Imagenette	2,340,000	3:15	90.69%	4.05E-03

B. Static Data Exclusion

We perform a total of six experiments for each dataset to show the effect of static data exclusion. Table II shows the experimental results of the Micro-PCB dataset, and table III shows the results of the Imagenette dataset.

The runtimes of all experiments are reduced; this is a result of fewer evaluations being performed, due to the data reduction. The amount of runtime reduction correlates proportionally to the amount of data exclusion used.

The percentage differences of the experiments shows a high amount of variance between lateral and central exclusion, which is an unexpected result. The expected behaviour is that the runtimes would be more similar, as the same amount of reduction is performed, and therefore the same amount of image evaluations. For example, with the Micro-PCB dataset, lateral exclusion of 1% of the dataset gives a runtime reduction of 4.280%, while central exclusion gives a runtime reduction of 8.422%. This amounts to 4 minutes, which could be the result of change in conditions that the computer was run under, such as increase in room temperature.

The results of the experimentation also show a decrease in average accuracy across all experiments. This decrease is also proportional to the amount of data excluded. However, there is a moderate difference in accuracy between central and lateral exclusion. For example, with the Micro-PCB dataset and data exclusion of 10%, lateral exclusion gives a decrease of 0.141% and central exclusion gives a decrease in 0.282%. We must consider the p-value of these experiments. Lateral exclusion has a p-value of 0.13, and central exclusion has 0.02. As it is more than 0.05, the values of lateral exclusion are shown to be less statistically significant than those of central exclusion, more prone to random change and less reliable. This may be reduced by modifying the model to exclude operations that use randomness, such as those in data augmentation.

The results of the Imagenette datasets show similar patterns. The maximum reduction in runtime is 8.034%, which is half of the maximum runtime reduction with the Micro-PCB dataset. This can be explained by considering that the image data of the Imagenette dataset is more complex than those of the Micro-PCB dataset, and as such, evaluating a single image takes more processing time.

The average accuracies of the Imagenette experiments show a pattern of lateral exclusion being outperformed by central exclusion. Firstly, all experiments, except lateral exclusion of 1%, have a p-value of less than 0.1. These low values indicate high statistical significance, which suggests a real difference in model performance. A large p-value would indicate low statistical significance, meaning any difference in model performance is likely down to random chance.

The accuracy reductions are:

- lateral exclusion of 5%: 1.024%
- central exclusion of 5%: 0.909%
- lateral exclusion of 10%: 1.895%
- central exclusion of 10%: 0.919%

The accuracy reductions of lateral exclusion experiments are worse then those of central exclusion. This correlates to work by [12], where data far from the classes' centroid appears to hold more important information for classifying an image.

C. Dynamic Data Exclusion

Based on work by [13], we split the training process into two sections. The first section uses a reduced dataset, and the second section uses the complete dataset. Figure 1 shows an example of the data step for the Micro-PCB dataset, where:

- $S_1^E = 125$ epochs,
- $S_2^E = 125$ epochs,
- $S_1^D = 90\%$ of the dataset,
- $S_2^D = 100\%$ of the dataset.

Table IV shows the experimental results on the Micro-PCB dataset, and table V shows the results on the Imagenette dataset.

The first notable result is that p-values are highly variable. The closer this value is to 1, the less statistically significant it

TABLE II: Micro-PCB dataset results with static data reduction

Test type	Evals	Data exclusion	Runtime (hours)	Test Accuracy		Percentage difference	
				Average	P-value	Runtime	Average Accuracy
Full	1,620,000	-	1:55	99.962%	-	-	-
Lateral exc.	1,590,000	1.00%	1:50	99.833%	0.1657	-4.280%	-0.128%
Central exc.	1,590,000	1.00%	1:46	99.897%	0.3015	-8.422%	-0.064%
Lateral exc.	1,530,000	5.00%	1:42	99.910%	0.4284	-11.781%	-0.051%
Central exc.	1,530,000	5.00%	1:45	99.731%	0.0712	-9.024%	-0.231%
Lateral exc.	1,440,000	10.00%	1:36	99.821%	0.1300	-16.654%	-0.141%
Central exc.	1,440,000	10.00%	1:36	99.679%	0.0210	-16.903%	-0.282%

TABLE III: Imagenette dataset results with static data reduction

Test type	Evals	Data exclusion	Runtime (hours)	Test Accuracy		Percentage difference	
				Average	P-value	Runtime	Average Accuracy
Full	2,340,000	-	3:19	90.686%	-	-	-
Lateral exc	2,310,000	1.00%	3:18	90.347%	0.3394	-0.962%	-0.373%
Central exc	2,310,000	1.00%	3:18	89.826%	0.0660	-0.996%	-0.948%
Lateral exc	2,220,000	5.00%	3:12	89.757%	0.0269	-3.548%	-1.024%
Central exc	2,220,000	5.00%	3:14	89.861%	0.0386	-2.989%	-0.909%
Lateral exc	2,100,000	10.00%	3:03	88.967%	0.0147	-8.034%	-1.895%
Central exc	2,100,000	10.00%	3:08	89.852%	0.0408	-5.910%	-0.919%

TABLE IV: Micro-PCB dataset results with dynamic data reduction

Test type	Evals	Data exclusion	Runtime (hours)	Test Accuracy		Percentage difference	
				Average	P-value	Runtime	Average Accuracy
Full	1,620,000	-	1:55	99.962%	-	-	-
Lateral exc.	1,605,000	1.00%	1:53	99.974%	0.7328	-2.150%	0.013%
Central exc.	1,605,000	1.00%	1:51	99.897%	0.3015	-3.381%	-0.064%
Lateral exc.	1,575,000	5.00%	1:53	99.974%	0.6810	-2.365%	0.013%
Central exc.	1,575,000	5.00%	1:50	99.885%	0.0598	-4.926%	-0.077%
Lateral exc.	1,530,000	10.00%	1:48	99.731%	0.2222	-6.427%	-0.231%
Central exc.	1,530,000	10.00%	1:48	99.910%	0.3393	-6.413%	-0.051%

TABLE V: Imagenette dataset results with dynamic data reduction

Test type	Evals	Data exclusion	Runtime (hours)	Test Accuracy		Percentage difference	
				Average	P-value	Runtime	Average Accuracy
Full	2,340,000	-	3:19	90.686%	-	-	-
Lateral exc.	2,325,000	1.00%	3:20	90.217%	0.1516	0.041%	-0.517%
Central exc.	2,325,000	1.00%	3:20	90.556%	0.7184	0.093%	-0.144%
Lateral exc.	2,280,000	5.00%	3:16	90.399%	0.3929	-1.744%	-0.316%
Central exc.	2,280,000	5.00%	3:11	90.122%	0.1394	-4.176%	-0.622%
Lateral exc.	2,220,000	10.00%	3:12	90.182%	0.4018	-3.609%	-0.555%
Central exc.	2,220,000	10.00%	3:12	90.512%	0.7091	-3.607%	-0.191%

should be considered as. This may seem like an unfavourable result, but this tells us that the results are less distinguishable from the benchmark accuracies. As we are trying to reduce the data and maintain performance, this is in fact ideal.

The results of the Micro-PCB dataset show two experiments with an increase in average accuracy. Lateral exclusion with both 1% and 5% exclusion have an accuracy increase of 0.013%. As previously mentioned, the high p-values of 0.74 and 0.68 show that the results are not statistically significant, and highly prone to small random changes in the network model. Therefore, unfortunately, we must conclude that this result is unreliable.

The results of the Imagenette dataset show average accuracy decrease across all experiments. As with the results of the Micro-PCB dataset, the results are unreliable due to the varying p-values. Any further analysis can only be considered as speculation.

V. Discussion

The results of the dynamic data reduction experiments give much higher p-values than those of static data reduction. This implies that the difference in accuracy compared to the benchmark is less statistically significant, and any differences may be down to random change. This means that the results

Citation information: DOI10.1109/PAINE56030.2022.10014786, 2022 IEEE Physical Assurance and Inspection of Electronics (PAINE)

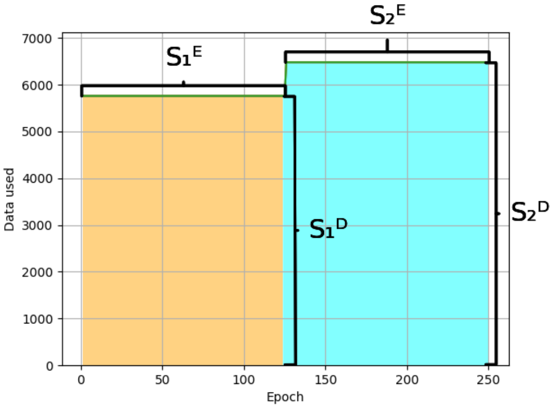

Fig. 1: Dynamic data allocation with a step from 90% data to full dataset use, for Micro-PCB dataset

are less distinguishable from the benchmark; as the task is to reduce the data use and maintain the accuracy as much as possible, this is a desirable result.

The runtime reductions of the Micro-PCB dataset are greater than those of the Imagenette dataset, despite the same amount of data reduction implemented. This result shows that the Imagenette dataset requires more time per image evaluation than that of the Micro-PCB dataset, which suggests that the images of the Micro-PCB dataset are less complex than those of Imagenette. Therefore, the Micro-PCB dataset can be classed as less computationally demanding, and more suitable for prototyping new technologies and further experimentation.

The Imagenette dataset has less performance reduction when trained using central data exclusion. This correlates to the conclusions of [12]. This means that the data furthest from the centroid holds more important features necessary for distinguishing the image from other classes, compared to data closer to the centroid.

VI. Conclusion

The results of this paper have shown the benefits of dynamic data reduction combined with data selection by means of lateral and central data identification. The combination of these techniques yields promising results, although there is a question regarding the reliability of the reported accuracies, due to the high p-values they are accompanied with. Future work might involve conducting experiments that yield low p-values, so there can be more certainty that the results are different to the benchmark. More extreme data reduction would likely force this difference.

Another research direction would be to experiment with alternative methods of dynamic data reduction, such as starting training with the full dataset and stepping down to a reduced dataset. Also, these methods may be applied to other datasets, to identify what type of data benifits these methods most. In terms of the Micro-PCB dataset, data may be selected based on its perspective, to give focus to the most valuable viewing angles.

The improved accuracies shown in this paper are small, but with further investigation and optimisation of dynamic data reduction, we can improve how we train our networks.

References

[1] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," *arXiv preprint arXiv:2007.05558*, 2020.

[2] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1328–1347, 2019.

[3] M. Hajij, G. Zamzmi, K. N. Ramamurthy, and A. G. Saenz, "Data-centric ai requires rethinking data notion," *arXiv preprint arXiv:2110.02491*, 2021.

[4] L. Chen, *Curse of Dimensionality*. Boston, MA: Springer US, 2009, pp. 545–546.

[5] A. Galdran, G. Carneiro, and M. Á. G. Ballester, "Balanced-mixup for highly imbalanced medical image classification," *CoRR*, vol. abs/2109.09850, 2021. [Online]. Available: https://arxiv.org/abs/2109.09850

[6] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[7] G. E. Hinton, P. Dayan, and M. Revow, "Modeling the manifolds of images of handwritten digits," *IEEE transactions on Neural Networks*, vol. 8, no. 1, pp. 65–74, 1997.

[8] A. Breger, M. Ehler, H. Bogunovic, S. Waldstein, A.-M. Philip, U. Schmidt-Erfurth, and B. Gerendas, "Supervised learning and dimension reduction techniques for quantification of retinal fluid in optical coherence tomography images," *Eye*, vol. 31, no. 8, pp. 1212–1220, 2017.

[9] S. Choi, J. H. Shin, J. Lee, P. Sheridan, and W. D. Lu, "Experimental demonstration of feature extraction and dimensionality reduction using memristor networks," *Nano letters*, vol. 17, no. 5, pp. 3113–3118, 2017.

[10] Y. Kiarashinejad, S. Abdollahramezani, and A. Adibi, "Deep learning approach based on dimensionality reduction for designing electromagnetic nanostructures," *npj Computational Materials*, vol. 6, no. 1, pp. 1–12, 2020.

[11] K. L. Turkiewicz, *Data Trimming*. SAGE Publications, Inc, 2017, pp. 347–348.

[12] A. Byerly and T. Kalganova, "Towards an analytical definition of sufficient data," *arXiv preprint arXiv:2202.03238*, 2022.

[13] D. Sanderson and T. Kalgonova, "Maintaining performance with less data," *arXiv preprint arXiv:2208.02007*, 2022.

[14] R. S. Raju, K. Daruwalla, and M. H. Lipasti, "Accelerating deep learning with dynamic data pruning," *CoRR*, vol. abs/2111.12621, 2021. [Online]. Available: https://arxiv.org/abs/2111.12621

[15] "Ml co2 impact," https://mlco2.github.io/impact/, accessed: 2022-08-03.