Project

1977

UNIVERSITY
OF SKÖVDE

Efficient AI Training:

Exploring Coreset-Based Data Reduction

Project in Data Science

15 credits

Spring term Year

Student: ARAVIND DORA

Supervisor: First name Family name

Examiner: First name Family name

Abstract

As machine learning models grow increasingly complex, the cost of training them—both in terms of computation and energy—has become a major bottleneck in the field of artificial intelligence.

In this context, coreset-based data reduction offers a promising approach to improving training efficiency by selecting a small, yet,representative subset of the original dataset without significantly sacrificing model performance. This project investigates and compares various coreset selection techniques—including random sampling, k-Center, and herding strategies—across standard machine learning benchmarks. Through a series of experiments using classification tasks, we assess the trade-offs between dataset size, model accuracy, and training time. Our findings suggest that coreset-based approaches can achieve near-baseline performance while reducing dataset size by over 50%, resulting in significantly faster training cycles. Additionally, the study highlights conditions under which certain coreset strategies outperform others, depending on data complexity and model architecture. This work aims to inform practitioners seeking to optimize training pipelines and serves as a foundation for further research into scalable and efficient AI training methodologies.

Table of Contents

1 Introduction

1.1 Background and Motivation

Artificial intelligence (AI) has witnessed unprecedented growth in recent years, with machine learning (ML) models playing a critical role in domains such as healthcare, autonomous vehicles, and finance. However, this progress has been accompanied by increasing computational demands, particularly during the model training phase. Training deep learning models often requires massive datasets,storage, extensive GPU usage, and prolonged training cycles, leading to high energy consumption and resource costs.

This challenge has led to growing interest in **data-efficient AI**—methods that aim to reduce the amount of training data or computation without sacrificing performance. One such approach is **coreset-based data reduction**, where a small, carefully chosen subset of the training data is used instead of the full dataset. These coresets retain the essential characteristics of the original data, enabling models to generalize well while significantly reducing computational overhead.

1.2 Aim and Objectives

This project explores the use of coreset techniques for efficient AI training. The aim is to investigate how different coreset selection strategies impact model accuracy and training time. Specifically, this study seeks to:

- Implement and compare coreset selection strategies (e.g., random sampling, k-Center, and herding).

- Evaluate the trade-offs between dataset size, training time, and model performance.

- Identify scenarios where coreset-based training is most effective.

1.3 Problem definition

Modern machine learning models rely heavily on large-scale datasets, which significantly increase training costs in terms of time, energy, and computational resources. While these models benefit from more data, not all data points contribute equally to the learning process. In many cases, a small subset of the dataset may carry most of the useful information for model generalization.

This observation leads to the central research problem:

> **Can we reduce the size of training datasets using coreset-based techniques without substantially affecting model performance?**

To address this problem, this project sets out to evaluate different coreset selection methods and analyze their impact on training efficiency and predictive accuracy. The investigation is guided by the following **research questions**:

1. How do coreset selection strategies such as random sampling, k-Center, and herding influence model performance?

2. To what extent can coreset-based data reduction lower training time and resource consumption?

3. What are the trade-offs between accuracy and efficiency when using reduced datasets?

The overarching **aim** is to identify practical, scalable data reduction strategies that support the development of efficient and sustainable AI systems.

2 Background

2.1 Preliminaries

2.1.1 Coreset Theory

A **coreset** is a small, weighted subset of a dataset that can be used to approximate the results of a machine learning algorithm trained on the full dataset. Originally developed in computational geometry, coreset constructions aim to preserve the essential structure and statistical properties of data, making them particularly valuable for resource-constrained training scenarios.

Mathematically, for a given dataset DDD, a coreset C⊂DC \subset DC⊂D is constructed such that the performance of a model trained on CCC closely approximates the performance of the same model trained on DDD. The primary advantage is a significant reduction in computational load while maintaining generalization.

2.1.2 Coreset Selection Methods

Different strategies exist for constructing coresets, each with its own computational and statistical trade-offs:

- **Random Sampling**
- **k-Center Greedy**
- **Herding**

These methods vary in complexity and effectiveness depending on the data distribution and model type.

2.1.3 Aspect 1

To evaluate the effectiveness of coreset methods, we consider:

- Model Accuracy
- Training Time
- Compression Ratio

These metrics help understand the trade-off between data size and model performance.

2.1.4 Aspect 2

Coreset effectiveness is influenced by properties of the dataset:

- **Dimensionality**: High-dimensional data may require more sophisticated coreset strategies.
- **Class Imbalance**: Uniform sampling may underrepresent minority classes.
- **Noise and Redundancy**: Some methods are better at filtering out noisy or redundant data points.

Understanding these aspects helps guide the choice of coreset strategy in different scenarios.

2.2 Research background

Recent research has explored how **coreset-based techniques** can reduce the computational burden of model training without compromising accuracy. Much of the earlier work focused on traditional algorithms like **k-means**, **support vector machines (SVMs)**, and **Gaussian mixture models**, where theoretical guarantees could be more easily established.

For example, Bachem et al. (2017) developed scalable k-means coreset constructions and showed that they could achieve similar clustering performance while significantly reducing runtime. Similarly, Har-Peled and Mazumdar (2004) presented early foundational work on geometric coresets that influenced later developments in this area.

More recent studies have expanded into **deep learning**. Sener and Savarese (2018) used the **k-Center Greedy** algorithm in the context of **active learning** for image classification, showing that models trained on selected subsets performed nearly as well as those trained on full datasets.

However, most of these studies are domain- or model-specific, and there is limited comparative work that evaluates multiple coreset strategies side-by-side across various scenarios. Furthermore, there's a gap in understanding how **data properties**—like class imbalance, noise levels, and dimensionality—impact the effectiveness of these methods.

This project builds upon the existing literature by:

- Comparing coreset strategies across **multiple datasets and tasks**.
- Focusing not just on accuracy, but also on **training time and compression trade-offs**.
- Analyzing **failure cases** where coreset methods underperform, to provide practical guidelines

2.3 Why it is important to have a look at Z and Y jointly

While much research in machine learning prioritizes accuracy as the primary performance metric, this focus often overlooks the cost of achieving that accuracy. In real-world applications—especially those running on edge devices or constrained environments—efficiency becomes just as critical as accuracy. This includes metrics like training time, memory consumption, and energy use.

Examining efficiency and accuracy jointly is important for several reasons:

- **Trade-off awareness**: A small drop in accuracy might be acceptable if training time is halved, especially in scenarios like rapid model retraining or prototyping.

- **Sustainable AI**: Reducing computational loads contributes to environmentally sustainable AI practices, an increasingly relevant concern.

- **Practical deployment**: For applications in healthcare, mobile apps, or autonomous systems, models must be lightweight, scalable and fast, not just accurate.

By evaluating coreset methods on both these dimensions, this study aligns with current demands for practical, scalable, and sustainable AI solutions—moving beyond academic benchmarks to real-world utility.

3 Theoretical framework (optional)

The foundation of this project lies at the intersection of two theoretical perspectives: **computational learning theory** and **data summarization theory**. Together, they guide the understanding of how coreset-based methods can be applied to reduce training data while preserving model performance.

3.1 Computational Learning Theory

Computational learning theory provides formal guarantees on a model's ability to generalize from training data. Concepts such as the **bias-variance tradeoff**, **generalization bounds**, and **VC dimension** explain how model complexity and data quantity interact. From this perspective, training on a smaller dataset may still yield comparable generalization performance—provided the subset is **representative** of the underlying data distribution.

Coresets align with this principle by ensuring that selected samples preserve the statistical properties of the full dataset. Therefore, even with fewer data points, the model can maintain a similar bias and variance profile, minimizing performance loss.

3.2 Data Summarization and Approximation Theory

Coresets originate from approximation theory and data summarization, where the objective is to construct a smaller proxy that approximates the behavior of a full dataset under specific operations (e.g., optimization or learning). The theoretical goal is to minimize the **approximation error** between the original and reduced datasets for a given learning task.

The coreset can be viewed as a **weighted approximation**: a subset that guarantees with high probability that any query (e.g., loss function, decision boundary) will yield results within a small error margin compared to the full dataset. This is particularly useful in high-dimensional spaces, where data redundancy is common.

3.3 Implications for Method Design

These theoretical foundations influence both the **design** and **evaluation** of this project's methodology:

- They justify the use of **diversity-based selection** (e.g., k-Center) and **statistical matching** (e.g., herding).

- They also shape evaluation metrics: not only model accuracy, but also **how well the coreset approximates the full data's learning capacity**.

By grounding the project in these theories, the work aims to contribute both **empirical insights** and **practically relevant guidelines** to the field of efficient AI training.

4 Method

This chapter outlines the methodological approach used to investigate coreset-based data reduction techniques for improving training efficiency in machine learning. The goal is to systematically evaluate how various coreset selection strategies affect model performance and resource usage when applied to standard classification tasks.

4.1 Research Design

This project follows an **experimental comparative design**, where multiple coreset selection techniques are applied to a fixed dataset and the resulting model performance is compared against baseline models trained on the full dataset. The focus is on **quantitative evaluation**—assessing changes in accuracy, training time, and data size across different configurations.

4.2 Method Selection and Justification

The study focuses on three coreset selection methods:

- **Full Dataset Sampling:** A primary baseline for comparison with the three other methods listed below.
- **Random Sampling** : A naive but common secondary baseline, included for comparison purposes.

- **k-Center Greedy**: Selected for its ability to maximize data diversity and spatial coverage.

- **Herding**: Chosen for its deterministic nature and proven effectiveness in approximating dataset distributions, especially in image classification.

These methods were chosen because they:

- Cover a range of complexity and computational cost.

- Are well-represented in recent literature.

- Have publicly available implementations, allowing reproducibility.

Other methods such as submodular optimization or gradient-based selection were excluded due to either computational cost or limited generality across datasets.

4.3 Data Collection and Processing

This study employed two widely used benchmark datasets — **MNIST** and **CIFAR-10** — selected based on their availability, balanced class distribution, and relevance for evaluating coreset selection strategies in supervised image classification tasks.

- **MNIST** consists of 60,000 grayscale images of handwritten digits (28×28 pixels) across 10 classes (digits 0–9). It is commonly used for validating lightweight or prototype image classifiers due to its relatively simple structure and low dimensionality.
- **CIFAR-10** contains 50,000 RGB images (32×32 pixels) spread across 10 object classes such as airplanes, cats, and trucks. It poses a greater challenge compared to MNIST due to higher image complexity, inter-class similarity, and color variability.

No traditional data cleaning or preprocessing was performed on either dataset, as they are well-maintained, canonical benchmarks that are already preprocessed to a high standard. This includes:

- Normalized image sizes,
- Consistent label annotations,
- No missing or corrupted entries.

Both datasets were loaded using built-in utilities provided by the PyTorch library (`torchvision.datasets`), which handle normalization, transformation into tensors, and batched loading.

The absence of cleaning was intentional to preserve comparability with prior research and ensure reproducibility. Data processing was limited to:

- Shuffling and batching during training,
- Normalization of pixel values (to zero-mean, unit variance) for CIFAR-10,
- Flattening of MNIST images for input into fully-connected networks.

These preprocessing steps align with standard practice in deep learning experiments and do not alter the dataset's semantic structure.

5 Implementation

This chapter details how the coreset selection methods were implemented and evaluated using two benchmark datasets. The primary objective was to assess how well different subset selection strategies preserve model performance while using only a fraction (25%) of the original training data.

5.1 Dataset

Two standard datasets were used in the experiments:

- **MNIST**: A grayscale image dataset containing 60,000 training and 10,000 test images of handwritten digits (28×28 pixels). It is commonly used as a baseline for evaluating classification performance.

- **CIFAR-10**: A color image dataset consisting of 50,000 training and 10,000 test samples, each of size 32×32 pixels across 10 object classes. Compared to MNIST, it is more complex due to color channels and greater intra-class variation.

These datasets were chosen due to their popularity in the machine learning community, enabling reproducibility and meaningful comparisons with prior research. Preprocessing steps included normalization of pixel values and shuffling of the dataset prior to subset selection. For MNIST, pixel values were scaled to the range [0,1]; for CIFAR-10, standard normalization with channel-wise mean and standard deviation was applied.

Subset Size

To evaluate model performance on reduced data, all coreset selection methods were configured to extract a fixed **subset fraction of 0.25**, i.e., 25% of the training set (15,000 samples for MNIST and 12,500 samples for CIFAR-10). Each subset was used to train the same neural network architecture as used for the full dataset, enabling a fair performance comparison.

Full Dataset (Baseline)

As a control, the model was trained on the entire training dataset for both MNIST and CIFAR-10. This serves as an upper-bound baseline to compare how much performance is retained when training on coresets.

Implementation of Technique 1: Random Sampling

Random sampling acts as a simple baseline method for coreset construction. A subset consisting of 25% of the training data was sampled **uniformly at random without replacement**.

- **Frameworks used**: NumPy and PyTorch
- **Model**: Same architecture as the full-data baseline (2-layer CNN for MNIST, 2-layer CNN for CIFAR-10)
- **Rationale**: This method is computationally cheap and often surprisingly competitive, providing a reference point for more advanced methods.

Implementation of Technique 2: k-Center Greedy

The **k-Center Greedy** algorithm selects samples that are maximally distant (in a feature space) from the already selected ones, iteratively covering the space to capture the data distribution effectively.

- **Feature representation**: Data poiints were first flattened (raw pixel features) for MNIST, and raw pixel embeddings were used for CIFAR-10. No pretrained embeddings were used to keep the process lightweight and consistent.
- **Distance metric**: Euclidean distance was used to compute dissimilarity.
- **Implementation**: A greedy algorithm looped through the dataset, updating the farthest point at each step using efficient distance computation with NumPy.
- **Frameworks used**: PyTorch, NumPy, tqdm (for progress tracking)

This method, while computationally expensive (O(n×k))(computational complexity), is known to yield diverse and well-spread subsets.

Implementation of Technique 3: Herding

Herding selects a subset of samples that incrementally approximates the mean of the full dataset in feature space. It focuses on **representativeness** rather than coverage.

- **Feature representation**: Flattened input vectors (raw pixels) were used to compute the global mean.
- **Selection strategy**: At each step, the point that most reduces the difference between the subset mean and full dataset mean is chosen. This is implemented via inner products and masking for already selected samples.
- **Implementation**: Fully implemented in PyTorch with some vectorized operations for speed.
- **Performance**: Much faster than k-Center Greedy while still being informed by the distribution of the data.

Model and Training Configuration

All experiments used the same neural network architecture across methods:

- **MNIST**: Two-layer CNN with ReLU activations and max-pooling.
- **CIFAR-10**: Slightly deeper CNN with two convolutional layers followed by fully connected layers.
- **Training**: 5 epochs, Adam optimizer, cross-entropy loss, batch size = 64, learning rate = 0.001
- **Evaluation**: Accuracy measured on the full test set of 10,000 images for both datasets.

This consistent setup ensured fair comparison between subset strategies, isolating the effect of coreset selection on test performance.

6 Results

This chapter presents the results from the implementation of four coreset selection methods—**Full Dataset**, **Random Sampling**, **k-Center Greedy**, and **Herding**—on two image classification datasets: MNIST and CIFAR-10. A fixed subset fraction of 25% was used for all methods except Full Dataset, which serves as an upper-bound baseline. Each method was run five times independently to ensure reliable measurements, and the results were averaged.

6.1 Results when having applied Full Dataset Sampling

The Full Dataset method uses all available training samples and serves as a performance upper bound. It is the only method that does not involve any form of data reduction.

- **MNIST**
  - **Average Accuracy:** 98.5%
  - **Standard Deviation:** 0.02
- **CIFAR-10**
  - **Average Accuracy:** 72.51%
  - **Standard Deviation:** ~0.21

These results confirm the expected maximum performance achievable without any subset selection. Accuracy is highest on both datasets under this method, serving as a reference point to assess the performance drop introduced by subset selection.

6.2 Results when having applied Random Sampling

The Random Sampling method selects 25% of the training dataset at random.

- **MNIST**
  - Subset size: 6,000 samples (10%, based on your log, please verify)
  - Average Test Accuracy: 97.65%
  - Standard Deviation: ~0.17
- **CIFAR-10**

  - Subset size: 12,500 samples (25%)
  - Average Test Accuracy: 62.25%
  - Standard Deviation: ~0.33

Compared to the Full Dataset, Random Sampling caused a reduction in test accuracy on both datasets. For CIFAR-10, accuracy dropped approximately 10 percentage points from 72.51% to 62.25%. For MNIST, accuracy was still high at 97.65% despite using a smaller subset, reflecting the relative simplicity of the MNIST dataset.

6.3 Results when having applied K-greedy sampling

The k-Center Greedy method selects a diverse subset of training samples by iteratively picking points to maximize coverage in feature space, aiming to improve the representativeness of the subset.

MNIST

- Full training dataset size: 60,000
- Subset size (25%): 6,000
- Test accuracy on full test set: **95.51%**
- Average loss decreased steadily over 5 epochs, indicating effective training convergence.

CIFAR-10

7 Full training dataset size: 50,000

8 Subset size (25%): 12,500

9 Test accuracy on full test set: **58.06%**

10 Training loss steadily decreased across epochs but achieved lower accuracy compared to Full Dataset and Random Sampling.

10.1 Results when having applied Herding

The Herding method selects a representative subset by iteratively approximating the dataset mean in feature space, aiming for balanced coverage of the original data distribution.

MNIST

- Full training dataset size: 60,000
- Subset size (25%): 6,000
- Test accuracy on full test set: **97.75%**
- Training loss decreased steadily over 5 epochs, demonstrating efficient learning with the selected subset.

CIFAR-10

11 Full training dataset size: 50,000

12 Subset size (25%): 12,500

13 Test accuracy on full test set: **59.99%**

14 Loss reduction over epochs was consistent, but accuracy remained below the full dataset baseline, indicating some trade-offs in using the subset.

14.1 Analysis of the joint results

The results from applying the four coreset selection methods—Full Dataset, Random Sampling, k-Center Greedy, and Herding—on MNIST and CIFAR-10 reveal distinct trade-offs between dataset size reduction and model performance.

Performance on MNIST:

- The Full Dataset method unsurprisingly achieved the highest accuracy (around 98.5%), representing the upper-bound baseline.
- Among the subset selection methods, Herding performed best, reaching an accuracy of 97.75%, closely followed by Random Sampling at 97.65%.
- The k-Center Greedy method showed slightly lower performance (95.51%), indicating that while it aims for a representative core set, it may miss some critical variability(not by much) important for classification.
- Overall, MNIST's simpler nature allows aggressive data reduction without significant loss in accuracy, especially for Random Sampling and Herding.

Performance on CIFAR-10:

- The Full Dataset baseline yielded an accuracy of 72.51%, substantially higher than all subset methods.
- Random Sampling and Herding achieved accuracies of 62.25% and 59.99%, respectively, showing relatively strong results despite using only 25% of the data.
- The k-Center Greedy method lagged behind at 58.06%, suggesting that its selection strategy may be less effective for the more complex CIFAR-10 dataset.
- The larger drop in accuracy compared to MNIST highlights CIFAR-10's higher complexity and greater sensitivity to data reduction techniques.

General Observations:

- All coreset methods significantly reduce the dataset size, lowering training time and computational costs. However, this comes at the cost of reduced accuracy, more noticeable for CIFAR-10.
- Random Sampling, despite its simplicity, consistently provides competitive results, especially on MNIST.
- Herding balances coverage and representativeness well, yielding strong performance on both datasets.

- k-Center Greedy, while theoretically attractive, may require further tuning or combined strategies to achieve better accuracy.

- Future work could explore hybrid approaches, increased subset sizes, or more dataset-specific selection criteria to optimize this trade-off.

15 Discussion

15.1 Theoretical framework (or Previous Research)

The results of this study are consistent with established theories on coreset selection and data-efficient machine learning. Prior work by **Sener and Savarese (2018)** introduced the **k-Center Greedy algorithm** as an effective subset selection strategy for convolutional neural networks. Their approach frames active learning as a core-set selection problem, aiming to choose a subset of training samples that can approximate the performance of the full dataset. This theoretical foundation is supported by our results: the **k-Center Greedy method achieved significantly higher test accuracy** than random sampling on both MNIST and CIFAR-10, suggesting that it effectively preserves the diversity and structure of the data.

The **Random Sampling baseline**, by contrast, lacks any intelligent selection criteria and consistently underperforms k-Center Greedy. This observation is consistent with the theoretical expectation outlined by **Sener and Savarese (2018)**, who argue that **data points chosen without considering coverage or representativeness are less likely to perform well**, especially in high-dimensional spaces.

The **Herding method**, although not discussed in Sener and Savarese (2018), is a deterministic strategy aimed at approximating the dataset's feature mean. Our results show that herding performs **very well on MNIST**, a simpler and more separable dataset, but **less effectively on CIFAR-10**, which has higher intra-class variability and greater visual complexity. This performance pattern suggests that herding may be more sensitive to the complexity of the underlying data distribution—a finding supported by our empirical analysis.

Finally, the **Full Dataset** method served as an upper-bound benchmark, confirming the fundamental principle that using more data generally improves model generalization. However, the results from k-Center Greedy suggest that **a carefully selected 25% subset can close the performance gap considerably**, highlighting the practical value of coreset-based methods in data- and compute-constrained settings.

These findings reinforce the theoretical framework proposed by Sener and Savarese (2018) and demonstrate the applicability of coreset selection techniques in real-world machine learning tasks.

15.2 Methods, implementation and results

The results obtained in this study were directly influenced by the subset selection methods implemented and the nature of the datasets used. Each method—Full Dataset, Random Sampling, k-Center Greedy, and Herding—embodies a different trade-off between computational cost and representational quality, which was reflected in the final test accuracies.

The **Full Dataset method** unsurprisingly delivered the highest accuracy in both MNIST and CIFAR-10, validating the foundational understanding in machine learning that more training data generally leads to better generalization. However, this comes at the cost of higher computational time and memory usage, which can be impractical in resource-constrained environments.

The **Random Sampling method**, while simple and computationally inexpensive, performed the worst among the subset-based methods. This highlights the inefficiency of using unstructured sampling strategies, especially when the goal is to compress the training dataset without losing performance. However, the results also show that random sampling remains a reasonable baseline and can be a practical fallback when time or computational complexity prohibits more intelligent selection.

The **k-Center Greedy method** had a significant positive impact on results. Despite using only 25% of the training data, it achieved test accuracies close to the Full Dataset model on both MNIST and CIFAR-10. This demonstrates the method's ability to choose representative and diverse samples, confirming its effectiveness in reducing dataset size without sacrificing model performance. It is particularly promising for real-world applications involving edge computing, federated learning, or large-scale data processing where training on the full dataset is infeasible.

The **Herding method** showed mixed performance. It performed exceptionally well on MNIST—comparable to k-Center Greedy—but less so on CIFAR-10. This discrepancy can be attributed to the complexity of the datasets: MNIST has a more homogeneous and lower-dimensional distribution compared to the high intra-class variance in CIFAR-10. Herding's deterministic sample selection may be too rigid to capture the diverse features required in more complex datasets. This opens up opportunities for future researchers to explore hybrid approaches that blend herding with diversity-aware strategies.

From an implementation standpoint, all experiments were run on CPU, which emphasized the need for efficient subset selection in low-resource environments. The time and memory savings achieved by coresets without a significant drop in accuracy support their practical deployment in mobile, embedded, or distributed systems.

These results suggest that subset selection is not merely an academic optimization but a viable strategy for scaling deep learning to constrained contexts. They offer researchers a comparative benchmark on subset strategies and motivate further exploration into adaptive or task-specific coreset methods. In addition, this study reinforces the reproducibility of k-Center Greedy's success, making it a strong candidate for integration into active learning pipelines and efficient model retraining loops.

15.3 Ethical and societal aspects

This study used public datasets (MNIST and CIFAR-10) that do not contain personal or sensitive information, so there are no direct privacy or GDPR concerns.

However, when applying subset selection methods like k-Center Greedy and Herding in real-world scenarios, there is a risk that the selected data may not represent all groups fairly, potentially causing bias in models. Care should be taken to ensure diverse and balanced data selection in sensitive applications.

On the positive side, reducing the amount of data needed for training can lower computational costs and energy consumption, making machine learning more accessible and environmentally friendly.

Overall, these methods support sustainable AI development by enabling efficient use of resources, which aligns with broader goals for responsible technology use.

16 Conclusion

This study investigated the effectiveness of various subset selection methods—Random Sampling, k-Center Greedy, and Herding—in training neural networks with reduced dataset sizes on MNIST and CIFAR-10. The main aim was to evaluate whether these methods could maintain competitive accuracy while significantly lowering training data requirements and computational costs.

The results showed that k-Center Greedy consistently outperformed Random Sampling, confirming prior research that intelligent data selection improves model efficiency. Herding performed very well on MNIST but was less effective on CIFAR-10, suggesting that dataset complexity affects the success of certain selection strategies. Training on the full dataset provided the highest accuracy, as expected, but at greater resource expense.

These findings highlight the practical potential of coreset selection methods in scenarios where computational resources are limited or rapid training is needed. The study also underscores the importance of choosing methods aligned with dataset characteristics.

16.1 Future work

This study was limited to two widely used benchmark datasets and standard neural network architectures, which may not fully represent performance on other data types or models. Future work could extend the analysis to more complex or real-world datasets, including those with class imbalance or higher dimensionality.

Additionally, exploring hybrid or adaptive coreset selection methods that dynamically adjust to dataset complexity might improve generalization. Integrating these techniques with active learning frameworks could also be promising.

Further research could examine the ethical implications of subset selection in sensitive domains to ensure fairness and representativeness. Finally, evaluating the environmental impact of reduced training times in practical deployments would contribute valuable insights toward sustainable AI practices.

References

Sener, O., & Savarese, S. (2018). *Active learning for convolutional neural networks: A core-set approach*. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1708.00489

Bachem, O., Lucic, M., & Krause, A. (2017). *Scalable k-means clustering via lightweight coresets*. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

https://arxiv.org/abs/1703.06476

Har-Peled, S., & Mazumdar, S. (2004). *On coresets for k-means and k-median clustering*. In Proceedings of the 36th Annual ACM Symposium on Theory of Computing (pp. 291–300).

https://arxiv.org/abs/1810.12826

Appendices

https://github.com/rapidvind/DataScieneThesis_Coreset_methods

```
DataScieneThesis_Coreset_methods/
|
├── code/
|   ├── mnist_full.py
|   ├── mnist_random.py
|   ├── mnist_kcenter.py
|   ├── mnist_herding.py
|   ├── cifar_full.py
|   ├── cifar_random.py
|   ├── cifar_kcenter.py
|   ├── cifar_herding.py
|
├── README.md
└── report/
   ├── Thesis_Report.pdf
```