When Few Words Do Trick*
Dataset Distillation for Large Language Models

Xindi Wu
Princeton University
xindiw@princeton.edu

Anirudh Ajith
Princeton University
aa8052@princeton.edu

Sabhya Chhabria
Princeton University
sc9625@princeton.edu

Abstract

There is a vast amount of textual data available from various sources such as websites, news articles, and academic manuscripts, and it can be easily accessed through datasets like the common crawl, which is almost 541TB in size. However, training large language models (LLMs) from scratch on these large datasets can be computationally infeasible in many settings. While there have been efforts to make LLM training more accessible, there has not yet been much exploration into the possibility of effectively distilling large-scale textual data as a solution. While we find that the dataset distillation approaches proposed in the vision community might be ineffective in the LLM setting, we hope that our work will help motivate 1) theoretical investigations into the minimum amount of information required to train performative LLMs, 2) investigations into understand the compositionality of the compact linguistic knowledge, 3) investigations into other more effective dataset distillation methods. More information can be found in https://github.com/XindiWu/LLM-Distillation/tree/llm.

1 Introduction

As language models continue to grow to unprecedented scales, it becomes increasingly important to focus on techniques that ameliorate the growing problem of the inaccessibility of these models in consumer settings. While various frameworks aimed at distilling trained LLMs themselves into smaller (yet performant) models have been studied extensively in the recent past, an area that has not received much attention in the LLM space is the problem of dataset distillation.

Recent LLMs have been trained on increasingly larger and larger datasets. Training on these enormous datasets can be prohibitively expensive in terms of both compute and storage not only on consumer devices, but even in many commercial or research settings. This issue motivates a new avenue for optimization - dataset distillation. Dataset distillation is a problem that aims to distill large amounts of data down to smaller (filtered or synthetically generated) datasets that can be used to train models which are ideally as performant as models trained using the original data.

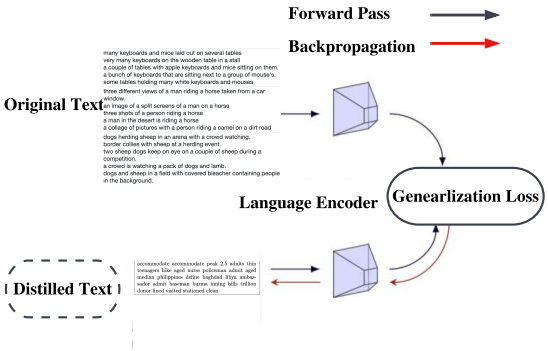

Figure 1: **Illustration of our proposed schema for LLM dataset distillation**. It includes two steps, the inner loop focuses on training the language model from scratch, the outer loop focuses on updating the distilled dataset.

Deep learning methods have made large strides in building task-specific models, but are shown to easily forget past knowledge when learning new tasks. One way to address this issue is to build models that can compress a large amount of information into a small memory storage space. The dataset distillation mechanism would allow the model to quickly retrieve memories from the past and recover the corresponding skills. Our goal is to understand the critical data information from dataset distillation perspective and aim to encode the information of optimized LLMs weights through a small synthetic dataset. We will further provide in-depth understanding of what would be the min-

[0]From *The office*: "Why Waste Time When Few Word Do Trick". https://www.youtube.com/watch?v=_K-L9uhsBLM

imum information for LLMs to learn in order to quickly recover a good performance. Fundamentally we are trying to understand how to decouple knowledge encoded in LLMs with dataset distillation approaches.

2 Related Work

The question of whether it is possible to generate synthetic examples that contain a high density of information and are further used to train models has been studied more so by the computer vision community. Dataset distillation was first proposed in (Wang et al., 2018) where the authors showed that training set of a few synthetic data points be used in conjunction with convolutional neural networks such as LeNet, AlexNet to achieve competitive performance on MNIST and CIFAR10 datasets.

Later, (Zhao et al., 2020; Zhao and Bilen, 2021; Cazenavette et al., 2022; Lee et al., 2022; Jiang et al., 2022) explored different ways of perform gradient or trajectory matching, many of which are studied in meta-learning approaches. Those approaches focused on matching the gradients that would arise when the model is trained using real data to those which arise when it is trained using synthetic data. Instead of matching the final weights of neural networks, another thread of work (Zhao et al., 2020; Wang et al., 2022; Lee et al., 2022) focuses on matching feature distributions of the real and synthetic data in the latent embedding space to better align features or preserve real-feature distribution. While most of these methods created synthetic datasets consiting of synthetic inputs paired with real labels, some more recent studies have also experimented with using soft trainable synthetic labels.(Sucholutsky and Schonlau, 2021; Bohdal et al., 2020).

Many of these works use a bilevel optimization framework to create synthetic data where an inner loop trains an initialized model to convergence using the current version of the synthetic dataset, while the outer loop goes on to evaluate the predictions of these trained models on real data and use the associated loss as a training signal to optimize the synthetic dataset. This bilevel optimization can be computationally intensive and there has been some work attempting to increase the efficiency of the framework. (Lorraine et al., 2020; Nguyen et al., 2020, 2021; Vicol et al., 2022; Zhou et al., 2022; Nguyen et al., 2020).

There is a vast amount of textual data available from various sources such as websites, news articles, and academic manuscripts, and it can be easily accessed through datasets like the common crawl, which is almost 541TB in size. However, training large language models (LLMs) from scratch on these large datasets has become increasingly expensive. While there have been efforts to make LLM training more accessible, there has not yet been much exploration into the possibility of effectively distilling large-scale textual data as a solution. One of the main challenges of distilling textual data is that it is inherently discrete, meaning that each word must belong to a limited vocabulary. In addition, there is a rich underlying structure to text, with sentences following fixed patterns according to grammar, and the context in which a piece of text is used can greatly affect its semantic interpretation. (Sucholutsky and Schonlau, 2021) use a latent-embedding method for condensing text data. Essentially, they avoid the discrete nature of optimization by performing distillation in a continuous embedding space. Specifically, they utilize a fixed text-encoder to learn continuous representations of each word in the condensed text and optimize them using the BPTT data-distillation framework proposed by (Wang et al., 2018). The condensed text representations are then decoded using a nearest-neighbor approach.

3 Method

In this section, we introduced the two dataset distillation frameworks we implemented, including (Deng and Russakovsky, 2022) and (Wang et al., 2018).

3.1 Problem Formulation

Consider a model f_θ which has been initialized using the weights $\theta_0 \sim \Theta$. Traditionally, given a training set $\mathbf{D} = \{x_i, y_i\}_{i=1}^N$ and a loss function ℓ, one may attempt to find the optimal weights

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \ell\left(f_\theta(x_i), y_i\right). \quad (1)$$

using some gradient-based optimization methods.

The problem of dataset distillation involves creating a synthetic dataset $\mathbf{S} = \{\hat{x}_i, \hat{y}_i\}_{i=1}^M$ such that $M \ll N$ which can be used to train a model (whose weights are initialized to $\phi_0 \sim \Theta$) to reach some final weights ϕ^* such that $f_{\theta^*} \approx f_{\phi^*}$ when evaluated over the input distribution.

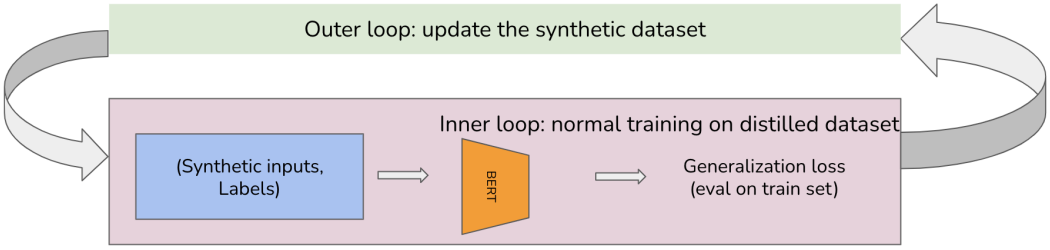

Figure 2: **Bilevel Optimization Structure.** Distilling a large-scale language dataset into compressed representation. A general dataset from $\mathcal{X}$ to $\mathcal{Y}$, can be distilled into these compressed representations for recall and model retraining.

Some formulations also return a sequence of step sizes $\mathbf{N} = \{\hat{\eta}_t\}_{t=1}^{T}$. These are to be used while performing T gradient-based updates of the model of the form

$$\phi_{t+1} = \phi_t - \hat{\eta}_t \nabla_\theta \left(\frac{1}{M} \sum_{j=1}^{M} \ell(f_\theta(\hat{x}_j), \hat{y}_j) \right) \Bigg|_{\theta=\phi_t} \tag{2}$$

This procedure described in 2 is referred to as the inner-loop optimization.

Finally, the model trained using the synthetic dataset f_{ϕ_T} is evaluated on the original training set $\mathbf{D}$ to obtain a "generalization" loss $\mathcal{L}(\mathbf{S}, \mathbf{N})$.

We can now compute the gradients of $\mathcal{L}$ with respect to $\mathbf{S}$ and $\mathbf{N}$ and perform updates

$$[\mathbf{S}_{\tau+1}, \mathbf{N}_{\tau+1}] = [\mathbf{S}_\tau, \mathbf{N}_\tau] - \alpha \nabla_{[\mathbf{S},\mathbf{N}]} \mathcal{L}(\mathbf{S}, \mathbf{N}) \Big|_{\substack{\mathbf{S}=\mathbf{S}_\tau \\ \mathbf{N}=\mathbf{N}_\tau}} \tag{3}$$

Equations 2 and 3 taken together comprise one iteration of the so-called outer loop optimization. By performing several iterations of the outer loop optimization, we can arrive at an $[\mathbf{S}, \mathbf{N}]$ which comprises the distilled dataset and the associated step size. These final values of $[\mathbf{S}, \mathbf{N}]$ can be used to train a model from scratch.

3.2 Implementation Details

In our case, f is a large language model. Owing to computational bottleneck concerns, we opted to use BERT-Tiny (Bhargava et al., 2021; Turc et al., 2019) while performing our data distillation experiments so far. However, we hope to extend our work to DistilBERT, BERT and RoBERTa Base as well in the future.

Also, since this formulation requires the synthetic dataset $\mathbf{S}$ to be differentiable, we fed token embeddings directly into the model during the inner-loop optimization rather than token IDs.

Although $\mathcal{L}$ is a function of $\mathbf{S}$ and $\mathbf{N}$, it is non-trivial to compute the gradients of $\mathcal{L}$ with respect to $[\mathbf{S}, \mathbf{N}]$ because of the multiple inner-loop iterations performed in 2 to train f from scratch. Backpropagation through time (BPTT) (Werbos, 1990) is an algorithm which is traditionally used to train recurrent neural networks on sequential data. It involves unrolling the RNN over a fixed number of time steps and using gradient descent to optimize the network's parameters.

After the inner-loop optimization in our dataset distillation framework 3, we use BPTT to calculate the gradients $\nabla_{[\mathbf{S},\mathbf{N}]} \mathcal{L}(\mathbf{S}, \mathbf{N})$.

Since we implement all our code using the PyTorch python library (Paszke et al., 2019), we can use `torch.autograd` module to perform the gradient computations internal to a single inner-loop iteration. However, owing to memory constraints, it is infeasible to carry out these gradient computations all the way from $\mathcal{L}$ to $[\mathbf{S}, \mathbf{N}]$. Because of this, we use the algorithms proposed by (Maclaurin et al., 2015; Domke, 2012) to carry out the full backpropagation for the outer loop optimization.

In practice, it is not always possible to perform full-batch gradient descent for equations 2 and 3 owing to memory constraints. Hence, we use batches of synthetic and real data drawn from $\mathbf{S}_\tau$ and $\mathbf{D}$ respectively in our implementation. We present our algorithm in 1.

4 Experiments

4.1 Evaluation Test-Bed

For choices of datasets to attempt to distill, we mainly consider the following four standard language datasets which are widely used in prior works on text classification.

Rotten Tomatoes: The Rotten Tomatoes (Pang

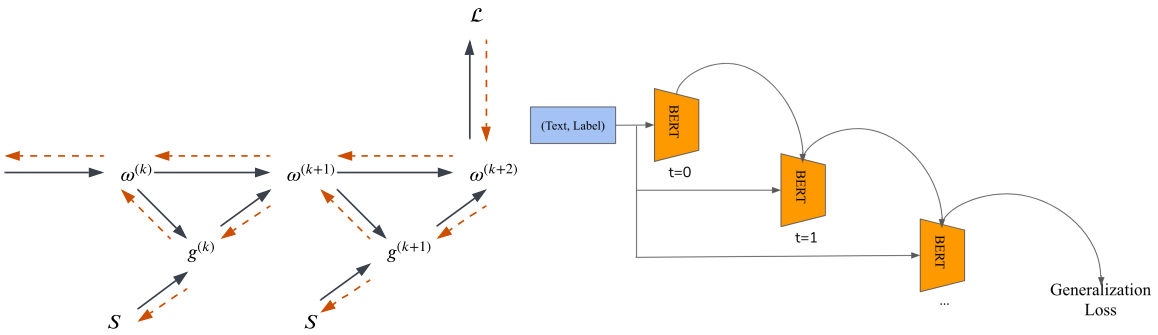

Figure 3: **Unrolled Optimization.** $\mathcal{L}$ is the training loss, in our case, we use the cross entropy loss to measure how accurately the text is classified to the right label. At step k, the model parameter and gradient are denoted as $\omega^{(k)}$ and $g^{(k)}$. The black solid line denotes a forward pass and the red dashed line denotes a backward pass. S here is the distilled dataset.

and Lee, 2005) dataset consists of movie reviews and ratings from the Rotten Tomatoes website, along with metadata about the movies. It includes a total of around 586,000 movie reviews, with around 290,000 labeled as positive and around 296,000 labeled as negative.

AG News: The AG News (Zhang et al., 2015) dataset consists of news articles from various sources, along with labels indicating the category of the article. It includes a total of around 120,000 news articles, with around 30,000 articles in each of the four categories (World, Sports, Business, and Science/Technology).

IMDB: The IMDB (Maas et al., 2011) dataset consists of movie reviews from the Internet Movie Database (IMDB). It includes a total of around 50,000 movie reviews, with around 25,000 labeled as positive and around 25,000 labeled as negative.

SST-2: The SST-2 (Socher et al., 2013) dataset is a subset of the Stanford Sentiment Treebank (SST) that consists of movie reviews annotated as either positive or negative. It includes a total of around 8,600 movie reviews, with around 4,300 labeled as positive and around 4,300 labeled as negative.

Implementation. We conduct all our experiments on a single GTX 3090 GPU with 24-GB VRAM. We train both our baseline models and our distillation framework using 100 epochs over the training data (for outer loop optimization in the case of the distillation experiments) with early stopping if convergence occurs before the 100th epoch. For the baseline experiments, we use an SGD optimizer and for the distillation experiments, we use Adam to optimize the synthetic inputs $\mathbf{S}$ and step sizes $\mathbf{N}$.

Model / Dataset	SST-2	RT	IMDB	AG News
TinyBERT	80.96	78.99	87.44	93.22
DistilBERT	80.74	78.71	87.32	92.97
BERT Base	93.35	84.71	93.86	94.46
RoBERTa Base	93.35	88.46	95.11	94.98

Table 1: **Baseline Results**: Here we report the baseline validation accuracies on four datasets with four different models.

	SST-2	RT	IMDB	AG News
TinyBERT	0.61	0.57	0.62	0.36
DistilBERT	-	-	-	-
BERT Base	-	-	-	-
RoBERTa Base	-	-	-	-

Table 2: **Distillation Results**: Here we report peak validation accuracy on four datasets while distilling via TinyBert. Note that these distillation runs were unsuccessful and our framework failed to converge.

5 Results

Using our experimental setup described above, we derive the following baseline results shown in Table 1 and the distilled experiments in Table 2. As we can see for each of the datasets, the baseline numbers and fairly high.

For each of the distillation experiments, we evaluated the model's accuracy (on the validation set of the corresponding dataset) when trained using $[\mathbf{S}, \mathbf{N}]$ after every epoch (outer loop).

Unfortunately, our dataset distillation pipeline remains very unstable and reliably fails to converge. We observed across various different settings of

Algorithm 1 Dataset Distillation for LLMs

hyperparameters: Momentum rate β_0, β_1, learning rate α_0, α_1 for θ and ϕ respectively.
input: Dataset $\mathbf{D}$, loss function $\ell(\cdot, \cdot)$
Initialize synthetic embeddings $\mathbf{S}$ and learning rates $\mathbf{N}$
repeat
  Randomly initialize model parameters θ_0
  Initialize momentum $\vec{m}_0 = 0$
  for $t = 1$ **to** T **do**
    Sample a minibatch $B_s = \{(xi', yi)\}$ from $\mathbf{S}$
    Compute $\mathcal{L} = \frac{1}{|B_s|} \sum_{i=1}^{|B_s|} \ell(f_{\theta_{t-1}}(xi'), y_i)$
    Update momentum $\vec{m}_t = \beta_0 \vec{m}_{t-1} + \frac{d\mathcal{L}}{d\vec{\theta}_{t-1}}$
    Update $\vec{\theta}_t = \vec{\theta}_{t-1} - \alpha_0 \vec{m}_t$
  end for
  Sample a minibatch $B = \{(xi, yi)\}$ from $\mathbf{D}$
  Compute $\mathcal{L}(\mathbf{S}, \mathbf{N}) = \frac{1}{|B|} \sum_{i=1}^{|B|} \ell(f_{\theta_T}(xi), yi)$
  Update $[\mathbf{S}, \mathbf{N}]$
until Converge

the hyperparameters of the distillation framework that any apparent increase in accuracy seemed to quickly be completely undone in the next epoch as the model reverted back to chance performance. We present a thorough analysis of why we believe this was the case in the following section.

6 Discussion

As mentioned above, while optimizing our distilled datasets, our models performed barely over chance when we experimented with distilling the 4 datasets mentioned in section 4.1. We remain confident in our theoretical understanding and implementation of the backpropagation through time algorithm and it is unclear where the pipeline breaks down. We experimented over a range of hyperparameters such as learning rate and scheduling, batch size, number of inner and outer loop optimizations, etc and we were still unable to achieve presentable results. We present a short analysis of a few possible reasons why we believe our methodology as described in section 3.

We hypothesize that the main reason for the poor performance of synthetic data in our LLM-based distillation pipeline is that transformer models (by virtue of their inductive biases) tend to model functions which are highly sensitive to their inputs. We believe that CNNs and previous language models like TextConvNet on which data distillation has successfully been shown, model mathematical functions which are more well-behaved in terms of Lipschitzness. The reason why we believe dataset distillation works so well for CNNs and RNNs is that smoothly changing the input results in a smooth change in the output as well. We believe this property does not hold to the same degree in transformers, wherein small perturbations to the input, cause the output change by a significant factor. We think that this makes it significantly harder to optimize $\mathbf{S}$ by backpropagating gradients through the entire sequence of inner-loop iterations.

7 Limitations

The "No Free Lunch" theorem (Wolpert and Macready, 1997) states that no single learning algorithm is guaranteed to be the best for all tasks. This also applies to dataset distillation, where the effectiveness of a data summary is influenced by the learning algorithm and objective function used during the distillation process. These choices represent encoded inductive biases, which can impact the ability of the summary to generalize to new data. It is important to consider these biases when selecting a dataset distillation technique, as they can significantly impact the resulting performance.

Previously, dataset distillation techniques have only been tested in situations where there is a very small amount of data (usually 1 to 50 data points per class). However, (Cui et al., 2022) found that when the size of the condensed data is increased, most distillation methods perform similarly to randomly sampling data. While this decrease in performance is expected, it occurs much more quickly with larger condensed data sets. Therefore, it is important to further study the reasons for this decline in performance and explore potential solutions if data distillation is to be used as a replacement for training with full datasets. If it proves to be impossible to address this limitation, there may be limited utility to data distillation frameworks as a whole.

One potential drawback of our current framework is that the inner optimization loop is very time-consuming, and it would be even more challenging to apply it to larger datasets. As we have

also reasoned in section 6, this bilevel framework makes optimization difficult. We intend to explore other learning frameworks, some methods that are known to ameliorate instability, and other closed-form solutions as a way to address this issue in our next steps.

8 Conclusion

8.1 Future Work

While our primary goal remains to work on mending our dataset distillation pipeline until we see positive results, there are many exciting ways in which we can extend this work. We discuss a few ideas below that we could potentially explore once we have a working distillation pipeline.

In-Context Learning: We hope to see if dataset distillation methods are applicable in the in-context learning setting. Our first move towards this goal will be to test whether the synthetic examples from distilled datasets can be used as viable prompts and compare the results prompts created using demonstrations randomly chosen from the original training set. In the text setting, distilled examples are represented my matrices in the embedding space. We hope to map these embeddings back to text by mapping each synthetic embedding to it's closest word embedding. After doing so, we would like to see if these synthetic text examples can make for viable prompts in the k-shot prompting setting. We believe that this might be a promising avenue because finding a way to distill the knowledge of the entire training set into a few synthetic demonstrations could potentially make for very potent prompts. These synthetic embeddings are likeley to be incoherent and not grammatically sound, so it would be interesting to see if they could yield similar levels of performance as data from the real dataset.

More Complex Tasks: Another important extension to our work would be to test this dataset distillation pipeline on a broader variety of tasks. We are especially curious to see how this pipeline would work with tasks that rely more on coherence and grammatical soundness of its inputs, for example, named entity recognition. We can also extend to more complex entailment and understanding-based tasks to see if the models can learn "more challenging" skills with smaller amounts of data.

Different Optimization Methods: Yet another way we would like to extend our work is to potentially explore optimization algorithms beyond stochastic gradient descent (for the inner loop optimization). The main reason we hope to try this is that SGD is no longer the state-of-the-art optimization algorithm used to train and finetune large language models. A natural extension to this work would be to use Adam to optimize the synthetic dataset. However, it remains unclear how we could do so.

8.2 Broader Impact

Improved efficiency: Distilling a large dataset into a smaller one can significantly improve the efficiency of the training process, making it faster and more practical to use in real-world applications. For the same number of gradient steps, dataset distillation can achieve much higher performance as opposed to optimizng on the standard dataset.

Increased accessibility: By making it easier to deploy and use large language models, data distillation can increase the accessibility of advanced techniques to a wider range of users and organizations.

Ethical considerations: Language data distillation raises a number of ethical considerations, such as the potential for bias in the data used to train the larger model and the potential for the smaller model to perpetuate that bias. It is important for practitioners to be aware of these issues and to take steps to address them. Another ethical consideration would be that dataset distillation could possibly make it easier for adversarial attackers to exploit models.

Economic implications: The development and use of large language models has significant economic implications, as these models have the potential to transform a wide range of industries. Data distillation can make it more feasible for organizations to adopt and use these models, which could drive innovation and economic growth while simultaneously minimizing environmental costs.

9 Acknowledgements

We thank Professor Danqi Chen, Alexander Wettig, as well as the other members of the COS597G course and Princeton NLP group for helping with proofreading and providing valuable feedback throughout.

References

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. 2020. Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. 2022. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759.

Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. 2022. Scaling up dataset distillation to imagenet-1k with constant memory. *arXiv preprint arXiv:2211.10586*.

Zhiwei Deng and Olga Russakovsky. 2022. Remember the past: Distilling datasets into addressable memories for neural networks. *arXiv preprint arXiv:2206.02916*.

Justin Domke. 2012. Generic methods for optimization-based modeling. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 318–326, La Palma, Canary Islands. PMLR.

Zixuan Jiang, Jiaqi Gu, Mingjie Liu, and David Z Pan. 2022. Delving into effective gradient matching for dataset condensation. *arXiv preprint arXiv:2208.00311*.

Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. 2022. Dataset condensation with contrastive signals. *arXiv preprint arXiv:2202.02916*.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. 2020. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Dougal Maclaurin, David Duvenaud, and Ryan P. Adams. 2015. Gradient-based hyperparameter optimization through reversible learning.

Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. 2020. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*.

Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. 2021. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ilia Sucholutsky and Matthias Schonlau. 2021. Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Paul Vicol, Jonathan P Lorraine, Fabian Pedregosa, David Duvenaud, and Roger B Grosse. 2022. On implicit bias in overparameterized bilevel optimization. In *International Conference on Machine Learning*, pages 22234–22259. PMLR.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. 2022. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12196–12205.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. 2018. Dataset distillation. *arXiv preprint arXiv:1811.10959*.

P.J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

David H Wolpert and William G Macready. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Bo Zhao and Hakan Bilen. 2021. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*.

Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. 2022. Dataset distillation using neural feature regression. *arXiv preprint arXiv:2206.00719*.