A Data Subset Selection Framework for Efficient Hyper-Parameter Tuning and Automatic Machine Learning

Savan Visalpara [1] **Krishnateja Killamsetty** [1] **Rishabh Iyer** [1]

Abstract

In recent years, deep learning models have found great success in various tasks viz., object detection, speech recognition, and translation, making the everyday lives of people easier. Despite the success, training a deep learning model is often challenging as its performance depends mainly on the hyperparameters used. Moreover, finding the best hyperparameter configuration is often time-consuming, even when using state-of-the-art (SOTA) hyper-parameter optimization algorithms as they require multiple training runs over the entire dataset for different possible sets of hyperparameters. Our main insight is that using a subset of the dataset for model training runs involved in hyper-parameter optimization allows us to find the optimal hyperparameter configuration significantly faster. In this work, we explore using the data subsets selected using the existing supervised learning based data subset selection methods, namely CRAIG, GLISTER, GRAD-MATCH, for model training runs involved in hyper-parameter optimization. Further, we empirically demonstrate through several experiments on real-world datasets that using data subsets for hyper-parameter optimization achieves significantly faster turnaround times for hyper-parameter selection that achieves comparable performance to the hyper-parameters found using the entire dataset.

1. Introduction

Modern deep learning and machine learning systems are shown to achieve near-human performance in many tasks. However, we are training the deep models with larger and larger datasets in the quest of near-human performance. Further, the performance of the machine learning and, in particular, deep learning models are dependent on the values of the hyper-parameters viz., learning algorithm, batch size, learning rate, model configuration parameters like depth, number of hidden layers, etc. Hence, it is becoming increasingly commonplace to run extensive hyper-parameter tuning and auto-ml pipelines to achieve state-of-the-art models. Though tuning the hyperparameters need multiple training runs over the entire datasets, resulting in staggering compute costs, running times, and, more importantly, CO2 emissions.

To be concrete, we consider an image classification task on a relatively simple CIFAR-10 dataset where a single training run using a relatively simple model class of Residual Networks on a V100 GPU takes around 6 hours. If we perform 1000 training runs with hyper-parameter tunings (which is not uncommon today), it will result in 6000 GPU hours. The resulting CO2 emissions would be between 640 to 1400 kg of CO2 emitted[1], which is equivalent to 1600 to 3500 miles of car travel in the US. The costs of training state-of-the-art NLP models and models on larger datasets like ImageNet are even more staggering (Strubell et al., 2019)[2].

This work will study the role of data subset selection for the simple task of hyper-parameter tuning and automatic machine learning. In particular, we will empirically study the role of different subset selection algorithms for the hyper-parameter tuning task and study the drop in accuracy with different data subsets and subset selection schemes. In particular, we will use three state-of-the-art data selection algorithms CRAIG (Mirzasoleiman et al., 2020), GLISTER (Killamsetty et al., 2021b), and GRAD-MATCH (Killamsetty et al., 2021a) and study the effect of these approaches on the hyper-parameter tuning task. So essentially, we will use subsets of data to tune the hyper-parameters. Once we obtain the tuned hyper-parameters, we will train the model (with the obtained hyper-params) on the full datasets. The smaller the data subset we use, the more the speedup and energy savings (and hence the decrease in CO2 emissions).

[*] Equal contribution [1] Department of Computer Science, University of Texas at Dallas, Dallas, USA. Correspondence to: Krishnateja Killamsetty <krishnateja.killamsetty@utdallas.edu>.

In the SubSetML Workshop 2021 at the International Conference on Machine Learning, Copyright 2021 by the author(s).

[1] https://mlco2.github.io/impact/#compute

[2] https://tinyurl.com/a66fexc7

1.1. Related Work

Data Subset Selection: A number of recent papers have used submodular functions[3] as *proxy* functions (Wei et al., 2014a;c; Kirchhoff & Bilmes, 2014; Kaushal et al., 2019). These approaches have been used in several domains, including speech recognition (Wei et al., 2014c;b), machine translation (Kirchhoff & Bilmes, 2014) and computer vision (Kaushal et al., 2019). Another common approach uses coresets. Coresets are weighted subsets of the data, which approximate certain desirable characteristics of the full data (*, e.g.*, the loss function) (Feldman, 2020). Coreset algorithms have been used for several problems including k-means and k-median clustering (Har-Peled & Mazumdar, 2004), SVMs (Clarkson, 2010) and Bayesian inference (Campbell & Broderick, 2018). Coreset algorithms require specialized (and often very different algorithms) depending on the model and problem at hand and have had limited success in deep learning. A very recent coreset algorithm called CRAIG (Mirzasoleiman et al., 2020) has shown promise for both deep learning and logistic regression models. Unlike other coreset techniques, which mostly focus on approximating loss functions, CRAIG tries to select representative subsets of the training data that closely approximate the full gradient. The resulting subset selection problem becomes an instance of facility location problem (which is submodular). Another data selection framework, which is very relevant to this work, poses the data selection problem as that of selecting a subset of the training data such that the resulting *model* (trained on the subset) performs well on the full dataset (Wei et al., 2015). One can also view this approach as obtaining a data subset that minimizes the KL divergence between the distribution induced by the parameters of the subset and that by the complete dataset. (Wei et al., 2015) showed that the resulting problem is a submodular optimization problem for Nearest Neighbor (NN) and Naive Bayes (NB) classifiers. In the NB case, the resulting function is the feature-based submodular function, whereas in the NN case, the function turns out to be the facility location function. Though the formulation holds only for NN and NB, these functions have also worked well for logistic regression and deep models (Kaushal et al., 2019; Wei et al., 2015).

Hyper-parameter tuning and auto-ml approaches: A number of algorithms have been proposed for hyper-parameter tuning including, grid search[4], bayesian algorithms (Bergstra et al., 2011), random search (Bergstra & Bengio, 2012), etc. Furthermore, a number of scalable toolkits and platforms for hyper-parameter tuning exist like Ray-tune (Liaw et al., 2018)[5], H2O automl (LeDell & Poirier, 2020), etc. See (Smith, 2018; Yu & Zhu, 2020) for a survey of current approaches and also tricks for hyper-parameter tuning for deep models. The biggest challenges of existing hyper-parameter tuning approaches are a) the large search space and high dimensionality of hyper-parameters and b) the increased training times of training models. Recent work (Li et al., 2018) has proposed an efficient approach for parallelizing hyper-parameter tuning using Asynchronous Successive Halving Algorithm (ASHA). This work is complementary to such approaches and could be combined effectively, though this is not something we look at in this work.

1.2. Contributions of the Work

To our knowledge, ours is the first work that studies the role of data subset selection for hyper-parameter tunings. In particular, we seek to ask a simple question: *can we use small data subsets of between 5% to 10% of the entire dataset, tune hyper-parameters and other aspects like model configurations on these much smaller subsets, and yet achieve comparable accuracies to tuning hyper-parameters on the full dataset?* We answer this question affirmatively by showing that, a) existing adaptive subset selection approaches like CRAIG, GRAD-MATCH, and GLISTER obtain hyper-parameters such that the model trained with these hyper-parameters are comparable in performance to the models trained with the hyperparameters obtained using entire dataset, and b) we show that these approaches significantly outperform random sampling approaches. We run experiments on three image classification and deep learning datasets, namely CIFAR-10, CIFAR-100, and MNIST.

2. DSS based HyperOpt and AutoML Framework

In this section, we will begin by presenting three data subset selection strategies that have been proposed in recent works.

2.1. CRAIG

The basic idea of CRAIG is to select a subset of data points such that the resulting subset of points has gradients that are representative of the entire dataset. To do this, the authors (Mirzasoleiman et al., 2020) selects a subset X which optimizes the facility location function (equivalent to the k-medoids) with the similarity function defined over the gradients. The resulting optimization problem is submodular and can be solved using a simple greedy algorithm (Nemhauser et al., 1978).

[3] Let $V = \{1, 2, \cdots, n\}$ denote a ground set of items. A set function $f : 2^V \to \mathbf{R}$ is a submodular (Fujishige, 2005) if it satisfies the diminishing returns property: for subsets $S, T \subseteq V, f(j|S)f(S \cup j) - f(S) \geq f(j|T)$.

[4] https://tinyurl.com/3hb2hans

[5] https://docs.ray.io/en/master/tune/index.html

2.2. GRAD-MATCH

The next approach, proposed very recently in (Killamsetty et al., 2021a) is called GRAD-MATCH. The approach is similar to CRAIG, but tries to directly minimize the gradient difference between the subset and the entire set. The optimization problem they study is:

$$w^t, X_t = \min_{w, X: |X| \leq k} \| \sum_{i \in X_t} w_i^t \nabla_\theta L_T^i(\theta_t) - \nabla_\theta L(\theta_t) \| \quad (1)$$

CRAIG (Mirzasoleiman et al., 2020) can be seen a minimizing the upper bound of this. In contrast, GRAD-MATCH directly optimizes equation grad-match, by optimizing the set function:

$$g(X) = \min_w \| \sum_{i \in X_t} w_i^t \nabla_\theta L_T^i(\theta_t) - \nabla_\theta L(\theta_t) \| \quad (2)$$

The authors (Killamsetty et al., 2021a) use a orthogonal matching pursuit based algorithm for this problem and also show that the resulting set function above (Equ. 2) is approximately submodular.

2.3. GLISTER

In this approach (Killamsetty et al., 2021b), the authors select a subset $X \subseteq V$ (i.e., select a subset of the training set), such that the resulting model θ_X performs well w.r.t a likelihood function LL_T on the entire dataset. Formally, we can write the optimization problem as:

$$\max_{X \subseteq V, |X| \leq k} LL_T(\text{argmax}_\theta \sum_{i \in X} LL_T^i(\theta)) \quad (3)$$

where LL_T^i refers to the likelihood of the ith training data point. A *dual* formulation of this problem is to find the minimum size subset X, such that $LL_T(\text{argmax}_\theta \sum_{i \in X} LL_T^i(\theta)) \approx LL_T(\text{argmax}_\theta \sum_{i \in V} LL^i(\theta))$. Past work (Wei et al., 2015) has studied this problem for a subclass of classifiers such as nearest neighbor and naive Bayes classifiers (Wei et al., 2015) for which the above optimization problem can be solved in closed-form. Additionally, this can also be solved in the case of linear regression and Gaussian naive Bayes classifiers, and interestingly, the resulting optimization problems turn out to be instances of submodular optimization problems (Wei et al., 2015). Unfortunately, the inner optimization problem, i.e. $\text{argmax}_\theta \sum_{i \in X} LL^i(\theta)$ does not admit a closed-form solution for more machine learning models such as logistic regression, SVMs, gradient boosted trees and multi-layer neural networks. Inspired by the progress made in solving bi-level optimization problems via meta-learning and one-step gradient approaches, (Killamsetty et al., 2021b) investigates an adaptive data selection approach, where they iteratively update the model parameters while also selecting the data subset. In particular, given the current parameters at epoch t as θ_t, we can select a subset X_t as:

$$X_t = \text{argmax}_{X \subseteq V, |X| \leq k} LL_T(\theta_t - \sum_{i \in X} \nabla LL_T^i(\theta_T)) \quad (4)$$

The above discrete optimization is submodular (under reasonable assumptions) for a number of loss functions such as cross-entropy, hinge loss, perceptron loss and logistic loss. As result, the optimization problem in equation one-step is a submodular maximization problem subject to cardinality constraint, for which efficient algorithms exist (Nemhauser et al., 1978). Similarly, the dual problem is an instance of submodular set cover (Wolsey, 1982).

2.4. Hyperparameter Tuning Algorithm

Any hyper-parameter tuning algorithm involves the following steps. First, there is a black box searching algorithm (which searches for the choices of hyper-parameters) either using Bayesian optimization, random search, or a simple grid search. The second is the model training which happens for every choice of the hyper-parameter. In this work, we propose to use data subset selection during the training. So essentially, given any hyper-parameter search algorithm, we can apply the data subset selection to speed up each training by a significant factor (say 10x), thereby significantly improving the experiment's overall turnaround times.

3. Experimental Results

Our experiments aim to demonstrate the stability and efficiency of using the subset selection approaches for hyper-parameter tuning. In most of our experiments, we study the tradeoffs between accuracy and efficiency (time/energy) of the different approaches presented above and compare it to Full training (which is a skyline in terms of accuracy) and random subset selection (which is a skyline in terms of time). In each of the subset selection approaches, we use warm start (Killamsetty et al., 2021a) which has shown to be very effective for data subset selection. To demonstrate the effectiveness of our approach, we performed experiments on CIFAR-100 (60000 instances) (Krizhevsky, 2009), MNIST (70000 instances) (LeCun et al., 2010), and CIFAR-10 (60000 instances) (Krizhevsky, 2009). Wherever the datasets do not have a pre-specified validation set, we split the original training set into a new train (90%) and validation sets (10%). All experiments were performed on V100 GPUs.

For hyper-parameter tuning, we use the Tree-structured Parzen Estimator (TPE) approach (Bergstra et al., 2011) though we believe the takeaways would remain the same even with other approaches.

Figure 1 shows the results on CIFAR-10 and CIFAR-100. For CIFAR-10, we use ResNet-18 as the model and set

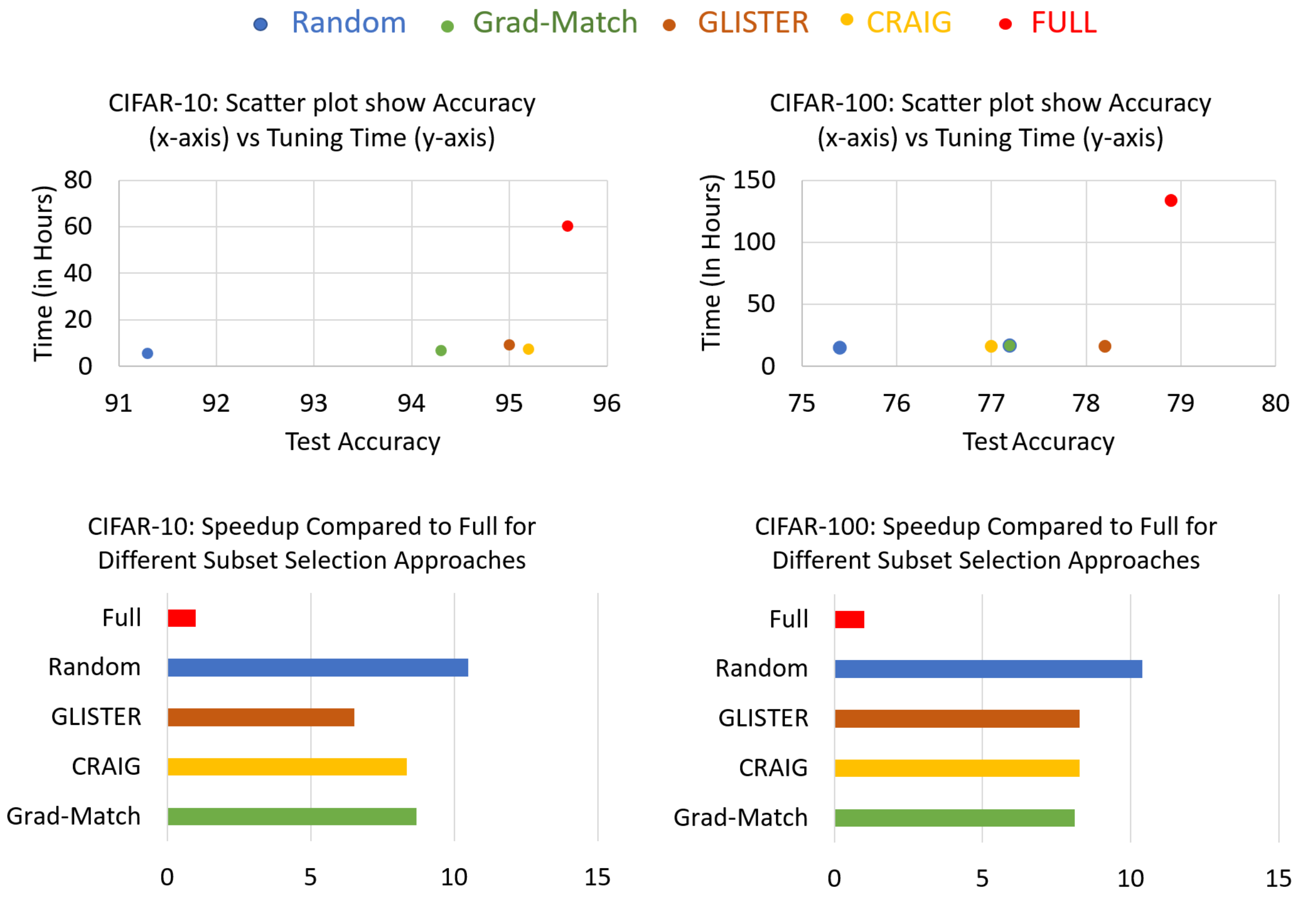

Figure 1. Top Figure: Comparing the training accuracy (from the selected hyper-parameters by the different tuning and selection approaches) on the x-axis and the end to end training time (which includes the hyper-parameter tuning) on the y-axis. We see that the subset selection approaches (CRAIG, GLISTER, GRAD-MATCH) are much faster than full training and also perform better than random sampling while being comparable in terms of running time. Bottom Figure shows the speedups of the different approaches above compared to full training. With a 5% subset, we see that the subset selection approaches achieve between 8x to 10x speedup compared to full training.

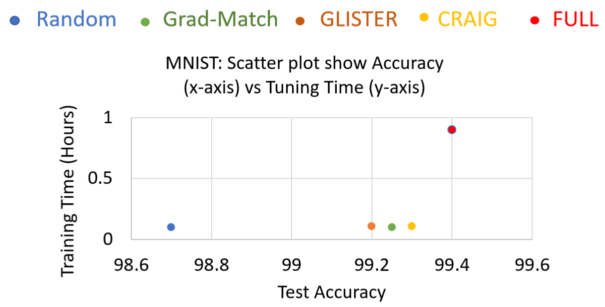

Figure 2. Comparison of Data selection approaches on MNIST. Again, we see that the subset selection achieves similar performance to full while enjoying 8x to 10x speedup and energy savings compared to full data.

the hyper-parameters as the learning rate, momentum, optimizer, number of epochs, and batch size. We set the number of hyper-parameter runs to be 10 in this experiment. For CIFAR-100, we use a ResNet-50 model and set the hyper-parameters as learning rate, momentum, optimizer, number of epochs, batch size, and the number of repetitions of bottleneck blocks model. In both cases, we use a 5% subset. We see that we achieve comparable performance to the entire dataset (slightly lower by 0.5%) while achieving significant speedup (around 10x speedup) by using the data subset selection. On the other hand, we see that the subset selection can obtain superior hyper-parameters than random sampling. Since we are using data subsets, the compute and energy savings from these approaches are also similar.

Figure 2 shows the results for MNIST. For MNIST, we used a LeNet model (LeCun et al., 1989). Again, we considered a search over the hyperparameters: learning rate, momentum, optimizer, number of epochs, and batch size. We see that the subset selection approaches can achieve within 0.05% of the full data performance while being close to 10x faster and energy-efficient.

4. Conclusion

To our knowledge, this is one of the first papers to study subset selection for hyper-parameter tuning. We study three state-of-the-art subset selection schemes and show that they achieve comparable performance to the entire dataset while enjoying close to 10x speedup. We see speedups consistently through different datasets (CIFAR-10, CIFAR-100, and MNIST) and different models (ResNet-18, ResNet-50, and LeNet). In future work, we would like to extend this work to consider a much richer set of hyper-parameters such as layer-wise learning rates, network architecture parameters, and problems like neural network architecture search, which are considerably more computationally expensive.

References

Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. In *25th annual conference on neural information processing systems (NIPS 2011)*, volume 24. Neural Information Processing Systems Foundation, 2011.

Campbell, T. and Broderick, T. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, pp. 698–706, 2018.

Clarkson, K. L. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):1–30, 2010.

Feldman, D. Core-sets: Updated survey. In *Sampling Techniques for Supervised or Unsupervised Tasks*, pp. 23–44. Springer, 2020.

Fujishige, S. *Submodular functions and optimization*. Elsevier, 2005.

Har-Peled, S. and Mazumdar, S. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 291–300, 2004.

Kaushal, V., Iyer, R., Kothawade, S., Mahadev, R., Doctor, K., and Ramakrishnan, G. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1289–1299. IEEE, 2019.

Killamsetty, K., Sivasubramanian, D., Mirzasoleiman, B., Ramakrishnan, G., De, A., and Iyer, R. Grad-match: A gradient matching based data subset selection for efficient learning. *In International Conference of Machine Learning, ICML*, 2021a.

Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., and Iyer, R. Glister: Generalization based data subset selection for efficient and robust learning. *In AAAI*, 2021b.

Kirchhoff, K. and Bilmes, J. Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 131–141, 2014.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

LeDell, E. and Poirier, S. H2o automl: Scalable automatic machine learning. In *Proceedings of the AutoML Workshop at ICML*, volume 2020, 2020.

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Hardt, M., Recht, B., and Talwalkar, A. A system for massively parallel hyperparameter tuning. *arXiv preprint arXiv:1810.05934*, 2018.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

Mirzasoleiman, B., Bilmes, J., and Leskovec, J. Coresets for data-efficient training of machine learning models, 2020.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Wei, K., Iyer, R., and Bilmes, J. Fast multi-stage submodular maximization. In *International conference on machine learning*, pp. 1494–1502. PMLR, 2014a.

Wei, K., Liu, Y., Kirchhoff, K., Bartels, C., and Bilmes, J. Submodular subset selection for large-scale speech training data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3311–3315. IEEE, 2014b.

Wei, K., Liu, Y., Kirchhoff, K., and Bilmes, J. Unsupervised submodular subset selection for speech data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4107–4111. IEEE, 2014c.

Wei, K., Iyer, R., and Bilmes, J. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pp. 1954–1963, 2015.

Wolsey, L. A. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4): 385–393, 1982.

Yu, T. and Zhu, H. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*, 2020.