Subset Selection with DeepHyper: Improving Hyperparameter Search with Subset Selection Strategies

Ian Wixom. Research Aide[1, a)] and Bethany Lusch. Mentor[1]
Argonne National Laboratory

(*Electronic mail: blusch@anl.gov.)

(Dated: 5 August 2022)

As the machine learning field continues to develop, the demand for more computational power and time has risen. The use of subset selection strategies in the context of hyperparameter searches is one potential approach in addressing the computational issue when defining an optimal model. We explored this approach by setting up four search scenarios with slightly differing dataset conditions, and then comparing the accuracy of each scenario's top performers. We found that subset selections can offer an up to 85.7% reduction in average search time with similar accuracy performance to full datasets.

I. INTRODUCTION

A. Subset Selection

Subset selection in the context of deep learning is a developing field with the objective of reducing training time with minimal loss in accuracy. The standard protocol for training neural network models is using the full training dataset. However, the energy required to use a full dataset can be, and is most likely, intensive. For example, training on a model class of Residual Networks with the CIFAR10 data set takes a staggering 6 hours on a V100 GPU[1]. Subset selection offers a potential solution to this issue by selecting a limited number of examples from the training data based upon certain strategies. Examples of such strategies include randomized subset selection, Coresets for Accelerating Incremental Gradient descent (CRAIG)[2] or Gradient Matching based Data Subset Selection (GradMatch)[3], the latter being a principle strategy of analysis for this report.

B. Hyperparameter Search

Hyperparameter search is another tool for deep learning that provides users the ability to search for high-performing hyperparameters for a given model. Software packages such as DeepHyper utilize Bayesian Optimization, a global optimizing strategy for black box scenarios, to perform hyperparameter searches[4]. Under the seminal publication on asynchronous model-based search methods (AMBS) from Prasanna Balaprakash, we see that DeepHyper's AMBS algorithm generates 50+% of its configurations above the 80% accuracy threshold, while other methods such as random search, genetic algorithms and hyperband have at most 30%[5].

C. Objective

The primary objective of this project is to see whether subset selection algorithms can help reduce time in searches while maintaining similar levels of accuracy compared to searches on a full training dataset. Four scenarios, two utilizing full datasets and two others utilizing subset algorithms, are considered during searches with limited amount of evaluations. Then, the top performing configurations from each scenario are trained to convergence to compare accuracy results.

II. PROGRESS

The project uses the Python programming language with a PyTorch library to construct and run neural network model training. To perform hyperparameter search (HPS) and data subset selection (DSS), the DeepHyper software package and CORDS library[6] are used respectively. To compare against the Grad Match results from the Decile Team, the project uses the CIFAR10 image database with a Residual Neural Network (ResNet), particularly ResNet-18, to serve as the model. The baseline uses the same default configuration from CORDS, and has hyperparameter variables for the learning rate in logarithmic distribution with the parameters of [0.00001, 0.5], regularization coefficient with a logarithmic distribution of [0.1, 10], momentum between [0.05, 0.95], a learning rate scheduler length between [1, 50] and weight decay between [0.00001, 0.005] with RMSprop as the optimizer.

There were four different scenarios that were evaluated. The first trained on a full dataset for the full number of epochs, and the second trained on a full dataset with half of the epochs. The third trained on a random subset of the full dataset at for the full number of epochs, and final scenario used Grad-Match to choose a smart subset of the training data for the full number of epochs. The total amount of configurations for each scenario was 50 and the batch size was 32 for all described scenarios. The Random and Grad-Match DataLoaders used 12.5% of the training data batch with no warm starting. More details about the dataloaders can be found by viewing the script itself at `https://github.com/1anw/DeepHyper-CORDS-Project`. Each search scenario

[a)] Also at Mahematics Department, University of Utah was ran with a full node from Theta-GPU, which comprises of 8 NVIDIA A100 GPUs for up to 3 hours using MPI to coordinate the search. After which, the top 8 configurations were ran for 100 epochs with a batch size of 128 on the full dataset for a fair comparison of each scenario's configurations. The top performing from each scenario was chosen as the best performing test accuracy. The scenarios generated the following results:

	avg. config run time	best performing
Full, 50	1465.27 sec.	92.5%
Full, 25	734.45 sec.	93.3%
Random	209.86 sec.	91.1%
Grad-Match	262.18 sec.	92.5%

We see that the accuracy after a more complete training is very similar with a range of 2.2%, with the full dataset with the half the number of total epochs performing the best. This was somewhat surprising as we hypothesized that Grad-Match would be the highest performing due to its optimized subsets. Nonetheless, Grad-Match and Random clearly outperformed the full dataset scenarios given the average time spent on an evaluation, with the 50 epoch and full dataset taking 24:25 per configuration run to Random's 3:29 which provides an 85.7% reduction for similar accuracy. Figure 1 below further demonstrates the time reduction compared to the full dataset with 50 epochs:

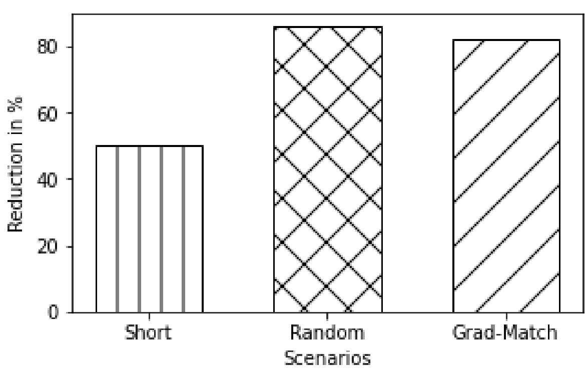

Figure 1. The time reduction for the scenarios compared to a full dataset was 49.9%, 85.7%, and 82.8% respectively.

III. FUTURE WORK

Based upon the results discussed in the Progress section, we see that subset selection strategies offer very high potential in hyperparameter searches. More research into other subset selection strategies that may be more fruitful in producing well-performing hyperparameters is needed. Furthermore, using different datasets to evaluate the search performance alongside different strategies would provide robustness to the potential claim of subset utility. Developing more generalized subset selection strategy programs for PyTorch, TensorFlow, and specialized dataloaders would be of high use as well. The potential scope required to accomplish these endeavors is relatively minimal.

IV. IMPACT ON LABORATORY

Deep learning has wide-reaching and impactful applications in grant-funded research both in the Laboratory and in its collaborators. As hyperparameter searches can be energy-intensive, utilizing random or smart subset algorithms has been proven to reduce the search time per configuration while achieving accurate results for the CIFAR10 dataset using ResNet architecture. This serves the interest of the laboratory's use of supercomputer resources, and the interest of researchers seeking improved model hyperparameters. We also are sharing tutorials explaining how to use CORDS and DeepHyper to do further research into subset selection and hyperparameter search.

V. CONCLUSION

The project explored the utility of subset selection strategies during hyperparameter searches. Under four scenarios, two that use the full dataset and two that use selection strategies, we found that the subset selection strategies performed at very similar levels to the full dataset with significant reductions in search time. While we expected the Grad-Match scenario to yield the highest accuracy, further exploration of other selection strategies is needed.

VI. ACKNOWLEDGEMENTS

This work was funded by and used resources of the Argonne Leadership Computing Facility (ALCF), a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. This work also used resources from Decile under MIT License 3.

[1] K. Killamsetty, G. S. Abhishek, A. V. Evfimievski, L. Popa, G. Ramakrishnan, R. Iyer, *et al.*, "Automata: Gradient based data subset selection for compute-efficient hyper-parameter tuning," arXiv preprint arXiv:2203.08212 (2022).

[2] B. Mirzasoleiman, J. Bilmes, and J. Leskovec, "Coresets for data-efficient training of machine learning models," in *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 119, edited by H. D. III and A. Singh (PMLR, 2020) pp. 6950–6960.

[3] K. Killamsetty, S. Durga, G. Ramakrishnan, A. De, and R. Iyer, "Grad-match: Gradient matching based data subset selection for efficient deep model training," in *International Conference on Machine Learning* (PMLR, 2021) pp. 5464–5474.

[4] M. S. V. V. S. W. Prasanna Balaprakash, Romain Egele, "DeepHyper: Scalable Asynchronous Neural Architecture and hyperparameter Search for Deep Neural Networks," (2022).

[5] P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath, and S. M. Wild, "Deephyper: Asynchronous hyperparameter search for deep neural networks," in *2018 IEEE 25th international conference on high performance computing (HiPC)* (IEEE, 2018) pp. 42–51.

[6] K. Killamsetty, D. Bhat, G. Ramakrishnan, and R. Iyer, "CORDS: COResets and Data Subset selection for Efficient Learning," (2022).