AutoML for Multilayer Perceptron and FPGA Co-design

1st Philip Colangelo
Intel PSG
San Jose, USA
philip.colangelo@intel.com

2nd Oren Segal
Hofstra University
Hempstead, USA
oren.segal@hofstra.edu

3rd Alex Speicher
Hofstra University
Hempstead, USA
aspeicher1@pride.hofstra.edu

4th Martin Margala
University of Massachusetts Lowell
Lowell, USA
Martin_Margala@uml.edu

2020 IEEE 33rd International System-on-Chip Conference (SOCC) | 978-1-7281-8746-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/SOCC49529.2020.9524785

I. Introduction

Optimizing neural network architectures (NNA) is a difficult process in part because of the vast number of hyperparameter combinations that exist. The difficulty in designing performant neural networks has brought a recent surge in interest in the automatic design and optimization of neural networks. The focus of the existing body of research has been on optimizing NNA for accuracy [1][2] with publications starting to address hardware optimizations [3]. Our focus is to close this gap by using evolutionary algorithms to search an entire design space, including NNA and reconfigurable hardware. Large data-centric companies such as Facebook[4][5] and Google [6] have published data showing that MLP workloads are the majority of their application base. Facebook cites the use of MLP for tasks such as determining which ads to display, which stories matter to see in a news feed, and which results to present from a search. Park et al. stress the importance of these networks and the current limitations on standard hardware and the call for what this research aims to solve, i.e., software and hardware co-design in [7]. Our research aims to take advantage of the reconfigurable architecture of an FPGA device that is capable of molding to a specific workload and neural network structure. Leveraging evolutionary algorithms to search the entire design space of both MLP and target hardware simultaneously, we find unique solutions that achieve both top accuracy and optimal hardware performance.

II. Approach: Evolutionary Cell Aided Design (ECAD)

The ECAD Evolutionary process, based on a steady-state model [8], generates a population of NNA/Hardware co-design candidates each with a complete set of parameters that effect both the accuracy and the hardware performance. The evolutionary search has three workers at its disposal to assess the fitness of various hardware platforms. The simulation worker is useful for assessing instruction-set based architectures such as CPU and GPU, whereas the physical and hardware database workers are useful for hardware that requires design and synthesis.

III. Experiments

We show the results from running a series of evolutionary searches on six different data sets: MNIST [9], Fashion MNIST [10], Credit-g [11], Har [12], Phishing [11], and Bioresponse [13]. Search was done on a Stratix 10 2800 FPGA and Titan X (Pascal) GPU.

Table I presents the top results obtained from the evolutionary algorithm searching for accuracy using k-fold cross-validation (note MNIST and Fashion MNIST use 1-fold). As can be seen, our mnist and fashion-mnist accuracy results outperform the top reported results. In addition our auto MLP network has the second best reported result and is 0.0047 shy of the SVC method record holder. Table II shows ECAD run time statistics for the results reported in Table I. It reports the number of different NNA/HW combinations that were automatically generated and evaluated by the ECAD system, the average time per evaluation and total evaluation time of all candidate architectures.

Table III shows the results for two top Pareto frontier solutions for each data set. The solutions provide accuracy and throughput for a Stratix 10 (S10) FPGA and TitanX (TX) GPU. In the majority of cases the FPGA achieved higher performance than the GPU. Credit-g, for example, favored GPU for higher accuracy, but looking at the second row for credit-g, by sacrificing just one point of accuracy, the FPGA sees a very significant improvement in throughput.

IV. Conclusions

We address the difficulty of designing highly performant neural networks by leveraging evolutionary search algorithms capable of finding the fittest solutions for both classification accuracy and hardware throughput. This process is shown to be both highly efficient and effective compared to traditional approaches that first design a neural network to achieve a target accuracy, then run it on general-purpose hardware. Through a series of experiments, we present our results for state of the art neural network configurations that surpass current published work. We explain the power of co-design by discussing the results of experiments showing accuracy versus throughput, performance scaling with bandwidth, and scaling designs with larger devices.

References

[1] Hanxiao Liu et al. “Hierarchical representations for efficient architecture search”. In: *arXiv preprint arXiv:1711.00436* (2017).

TABLE I: Top Accuracy (Acc) for All Datasets Compared to Previous Works

Dataset	Top Acc (Any)	Top Method	Top Acc (MLP)	MLP Type	ECAD MLP	K-fold
Credit-g	0.7860	mlr.classif.ranger	0.7470	*MLPClassifier	**0.7880**	10
Har	0.9957	*DecisionTreeClassifier	0.1888	*MLPClassifier	0.9909	10
Phishing	0.9753	*SVC	0.9733	*MLPClassifier	**0.9756**	10
Bioresponse	0.8160	mlr.classif.ranger	0.5423	*MLPClassifier	0.8038	10
MNIST	0.9979	Manual	0.9840	Manual(no distortions)	0.9852	1
Fashion MNIST	0.8970	SVC	0.8770	MLPClassifier	0.8923	1

Note The OpenML datasets/results can be found at openml.org: credit-g(https://www.openml.org/t/31), har(https://www.openml.org/t/14970), Phishing(https://www.openml.org/t/34537) and Bioresponse(https://www.openml.org/t/14966). Entries with * denote models from sklearn.

TABLE II: Top Accuracy Run Time Statistics

Dataset	Total Models Evaluated	AVG Model Evaluation Time (s)	Total Evaluation Time (s)
MNIST	553	71.23	39388.6
Fashion MNIST	481	82.55	39708.7
Credit-g	10480	2.24	23495.2
Har	3229	10.20	33069.4
Phishing	3534	9.24	32661.3
Bioresponse	5309	5.89	31285.0

Note Each model generated is a fully functional combination of NNA traits and hardware traits that is evaluated for performance on any of the measured metrics. The ECAD system *caches* similar configurations and avoids reevaluating them.

TABLE III: Best Pareto Frontier Results for Searching Accuracy and Throughput

Dataset	Accuracy	S10 (output/s)	TX (output/s)
MNIST	0.9841	7.97E5	7.73E5
MNIST	0.9763	2.45E6	1.97E6
Fashion MNIST	0.893	4.8E5	8.1E5
Fashion MNIST	0.8850	1.92E6	2.3E6
Har	0.996	1.16E6	9.59E5
Har	0.985	4.74E6	2.46E6
Credit-g	0.83	8.19E3	1.59E6
Credit-g	0.82	1.40E7	1.23E6
Bioresponse	0.798	4.64E5	1.34E6
Bioresponse	0.7952	1.36E6	1.66E6
Phishing	0.9675	6.81E6	2.27E6
Phishing	0.9656	1.16E7	2.27E6

[2] Esteban Real et al. "Large-scale evolution of image classifiers". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 2902–2911.

[3] Weiwen Jiang et al. "Hardware/Software Co-Exploration of Neural Architectures". In: *CoRR* abs/1907.04650 (2019). arXiv: 1907 . 04650. URL: http://arxiv.org/abs/1907.04650.

[4] Kim Hazelwood et al. "Applied machine learning at facebook: A datacenter infrastructure perspective". In: *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2018, pp. 620–629.

[5] Carole-Jean Wu et al. "Machine learning at facebook: Understanding inference at the edge". In: *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2019, pp. 331–344.

[6] Norman Jouppi et al. "Motivation for and Evaluation of the First Tensor Processing Unit". In: *IEEE Micro* 38.3 (2018), pp. 10–19.

[7] Jongsoo Park et al. "Deep learning inference in facebook data centers: Characterization, performance optimizations and hardware implications". In: *arXiv preprint arXiv:1811.09886* (2018).

[8] David E Goldberg and Kalyanmoy Deb. "A comparative analysis of selection schemes used in genetic algorithms". In: *Foundations of genetic algorithms*. Vol. 1. Elsevier, 1991, pp. 69–93.

[9] Yann LeCun and Corinna Cortes. "MNIST handwritten digit database". In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[10] Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms". In: *CoRR* abs/1708.07747 (2017). arXiv: 1708.07747. URL: http://arxiv.org/abs/1708.07747.

[11] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[12] Davide Anguita et al. "A public domain dataset for human activity recognition using smartphones." In: *Esann*. 2013.

[13] Boehringer Ingelheim. 2011. URL: https://www.openml.org/d/4134.