Latin Hypercube Initialization Strategy for Design Space Exploration of Deep Neural Network Architectures

Heitor R. Medeiros
Federal University of Pernambuco
Center for Strategic Technologies of the Northeast
Recife, Pernambuco
hrm@cin.ufpe.br

Diogo M. F. Izidio
Federal University of Pernambuco
Center for Strategic Technologies of the Northeast
Recife, Pernambuco
diogoizidio@hotmail.com

Antonyus P. do A. Ferreira
Center for Strategic Technologies of the Northeast
Recife, Pernambuco
pyetrotype@gmail.com

Edna N. da S. Barros
Federal University of Pernambuco
Recife, Pernambuco
ensb@cin.ufpe.br

ABSTRACT

In recent decades, deep learning approaches have shown impressive results in many applications. However, most of these approaches rely on manually crafted architectures for a specific task in large design space, allowing room for sub-optimal designs, which are more prone to be stuck in local minima and to overfit. Therefore, there is considerable motivation in performing architecture search for solving a specific task. In this work, we propose an initialization technique for design space exploration of deep neural networks architectures based on Latin Hypercube Sampling (LHS). When compared with random initialization using standard datasets in machine learning such as MNIST, and CIFAR-10, the proposed approach shows to be promissory on the neural architectural search domain, outperforming the commonly used random initialization.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; **Genetic algorithms**;

KEYWORDS

latin hypercube, initialization strategies, architecture search, deep learning

ACM Reference Format:
Heitor R. Medeiros, Diogo M. F. Izidio, Antonyus P. do A. Ferreira, and Edna N. da S. Barros. 2019. Latin Hypercube Initialization Strategy for Design Space Exploration of Deep Neural Network Architectures. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321922

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321922

1 INTRODUCTION

Evolutionary approaches for the design of deep learning architectures are often computationally demanding. Therefore, reducing the number of evaluations necessary for achieving an optimal solution is essential for allowing wide usage of these approaches in practical applications. Many works have tried to improve convergence with modifications in the algorithm such as crossover and mutation methods, selection mechanisms and adaptive controlling of parameter settings [3]. However, even though there is little research in this field, initialization techniques can often improve the convergence time as well as the quality of the final solution by guaranteeing diversity in the initial population [6]. The problem with random initialization is that often individuals are not evenly distributed throughout the search region. Therefore, the evolutionary search is more prone to get stuck in a local optimum. In this work, we propose the usage of Latin Hypercube Sampling (LHS) [7] initialization technique for design space exploration in the deep learning context. The proposed method shows to be promissory on the Neural Architectural Search (NAS) domain, outperforming the commonly used random initialization.

2 EVOLUTIONARY TECHNIQUE FOR DESIGN SPACE EXPLORATION

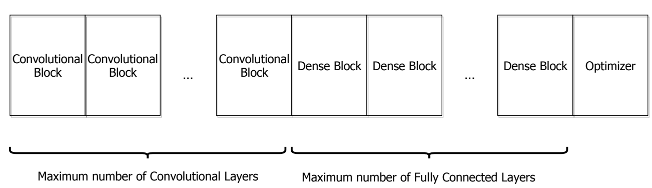

Figure 1: Final architecture for the neural network based on convolutional and dense blocks.

The genome used is a fixed-length vector based on Davison [1] work, which can be visualized in Figure 1. Each block encodes a layer in the final architecture which can be activated or deactivated by an boolean parameter, and contains information about the number of elements, the presence of Batch Normalization, the activation function, and the dropout rate for the current layer from a finite set of possible values. The convolution block differs by also including a boolean element which allows a 2x2 MaxPooling operation after the layer. Therefore, the architecture used as a solution is constructed by stacking these two kinds of blocks followed by an optimizer block, which determines the optimizer used for training. By using integers to represent the possible values for each parameter, an evolutionary search can be performed by using LHS [7] as an initalization technique.

3 EXPERIMENTS

This work was developed using Distributed Evolutionary Algorithms in Python (DEAP) [2] on MNIST [5] and CIFAR-10[4] datasets. We evaluated two scenarios: first, a comparison between the LHS approach and the random initialization on CIFAR-10; and second, using a larger search space to find the best model with LHS initialization over CIFAR-10 and MNIST. The parameters for each experiment are shown in Table 1.

Hyperparameter Name	First Experiment	Second Experiment
Number of Generations	20	30
Population Size	5	30
Max. Training Epochs	30	30
Max Convolutional Layers	4	6
Max Dense Layers	2	4
Max Conv. Filters p/ Layer	16	256
Max Dense Nodes p/ Layer	32	128
Crossover Rate	0.5	0.5
Genome Mutation Rate	0.2	0.3
Gene Mutation Rate	0.05	0.05

Table 1: Hyperparameter values from first and second experiment.

4 RESULTS AND CONCLUSIONS

The results obtained from the experiments without any preprocessing, leverage our approach to be a promissory feature for NAS domain. In Figure 2, thirty (30) runs for each type of sampling method (random and LHS) were performed, obtaining the mean and standard deviation for each generation. The experiment shows that on CIFAR-10, LHS outperforms the random sampling over the same conditions. On the second scenario, we were searching for a competitive result in common deep learning datasets, so we performed a larger space search. In Figures 3 and 4, a total of 900 (30 population and 30 generations) models were trained for each dataset with an early stop based on validation accuracy to prevent overfitting, reaching the mean of 93.58% and 59.94% on MNIST and CIFAR-10, respectively. The best generations were the 19th with a mean of 97.46% for MNIST and 13th with a mean of 64.11% for CIFAR-10.

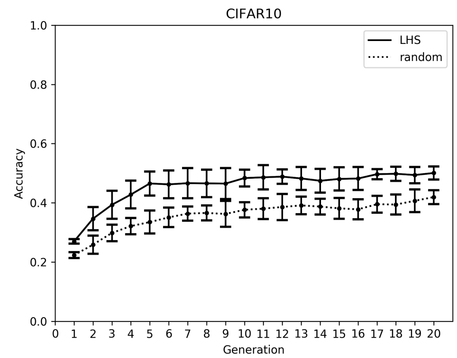

Figure 2: The first scenario, the central point represents the mean, and the vertical line is the standard deviation on CIFAR-10.

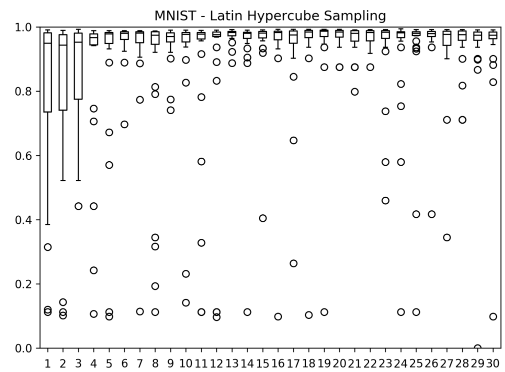

Figure 3: The second scenario on MNIST with LHS. The boxplot shows the median and interquartile range. The circles represent outliers.

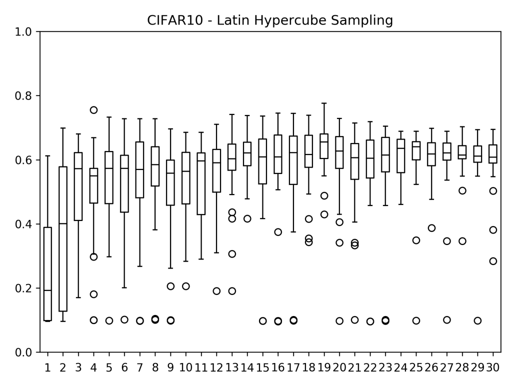

Figure 4: The second scenario on CIFAR-10 with LHS. The boxplot shows the median and interquartile range. The circles represent outliers.

REFERENCES

[1] J. Davison. 2017. DEvol - Deep Neural Network Evolution. https://github.com/joeddav/devol/. (2017).

[2] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research* 13, Jul (2012), 2171–2175.

[3] Erfan Khaji and Amin Satlikh Mohammadi. 2014. A Heuristic Method to Generate Better Initial Population for Evolutionary Methods. *arXiv preprint arXiv:1406.4518* (2014).

[4] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images.* Technical Report. Citeseer.

[5] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[6] Heikki Maaranen, Kaisa Miettinen, and Marko M Mäkelä. 2004. Quasi-random initial population for genetic algorithms. *Computers & Mathematics with Applications* 47, 12 (2004), 1885–1895.

[7] Michael D McKay, Richard J Beckman, and William J Conover. 2000. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 42, 1 (2000), 55–61.