Large Scale Evolution of Convolutional Neural Networks Using Volunteer Computing

Travis Desell
University of North Dakota
3950 Campus Road, Grand Forks, North Dakota 58201
tdesell@cs.und.edu

ABSTRACT

This work presents a new algorithm called evolutionary exploration of augmenting convolutional topologies (EXACT), which is capable of evolving the structure of convolutional neural networks (CNNs). EXACT is in part modeled after the neuroevolution of augmenting topologies (NEAT) algorithm, with notable exceptions to allow it to scale to large scale distributed computing environments and evolve networks with convolutional filters. In addition to multithreaded and MPI versions, EXACT has been implemented as part of a BOINC volunteer computing project, allowing large scale evolution. During a period of two months, over 4,500 volunteered computers on the Citizen Science Grid trained over 120,000 CNNs and evolved networks reaching 98.32% test data accuracy on the MNIST handwritten digits dataset. These results are even stronger as the backpropagation strategy used to train the CNNs was fairly rudimentary (ReLU units, L2 regularization and Nesterov momentum) and these were initial test runs done without refinement of the backpropagation hyperparameters. Further, the EXACT evolutionary strategy is independent of the method used to train the CNNs, so they could be further improved by advanced techniques like elastic distortions, pretraining and dropout. The evolved networks are also quite interesting, showing "organic" structures and significant differences from standard human designed architectures.

CCS CONCEPTS

•Computing methodologies → Discrete space search; Randomized search; Object identification;

KEYWORDS

Neuroevolution, Convolutional Neural Networks, Image Classification, Distributed Evolutionary Algorithms

1 EXACT

The EXACT algorithm starts with the observation that any two filters of any size within a CNN can be connected by a convolution of size $conv_d = |out_d - in_d| + 1$, where out_d and in_d are the size of the output and input filters, respectively, and $conv_d$ is the size of the convolution in dimension d. The consequence of this observation is that the structure of a CNN can be evolved solely by determining the sizes of the filters and how they are connected. Instead of evolving the weights of individual neurons and how they are connected, as done in the NEAT [9] algorithm, the architecture of a CNN can be evolved in a similar fashion except on the level of how filters are connected, with additional operators to modify the filter sizes.

Generation of the initial population starts with first generating a *minimal CNN genome*, which consists solely of the input node, which is the size of the training images (plus padding if desired), and one output node per training class for a softmax output layer. In this case of this work which uses the MNIST handwritten digits dataset, this is a 28x28 input node, and 10 output nodes. This is sent as the CNN genome for the first work request, and also inserted into the population with ∞ as fitness, denoting that it had not been evaluated yet. Further work requests are fulfilled by taking a random member of the population (which will be initially just the minimal CNN genome), mutating it, inserting the mutation into the population with ∞ as fitness and sending that CNN genome to the worker to evaluate. Once the population has reached a user specified population size through inserting newly generated mutations and results received by workers, work requests are fulfilled by either mutation or crossover, depending on a user specified crossover rate (*e.g.*, a 20% crossover rate will result in 80% mutation). Mutation operations include adding adges, enabling edges, disabling edges, splitting edges, and change node sizes.

2 EVOLVED NEURAL NETWORKS

Two simultaneous EXACT searches were performed, one using a epigenetic weight initialization where weights were reused from a previously trained parent, and the other using randomized weight initialization. Table 2 shows the error and prediction rates for the top 20 CNNs in these searches after 60,000 evaluated CNNs each. Table 1 shows the error and prediction rates for benchmark CNNs trained with the same hyperparameters. Figure 1 shows an example evolved CNN which had the highest test accuracy. These evolved networks are quite interesting in that they are highly different from the highly structured CNNs found seen in literature [5, 4, 8, 10, 1]. Even so, compared to hand designed benchmark networks, they still were able to find CNNs that trained to significantly lower training and test error, while making strong improvements in training and testing accuracy. The networks also show vestigial filters and edges, resulting from the crossover and edge disable mutation operator.

3 DISCUSSION AND FUTURE WORK

A novel algorithm for the evolution of arbitrarily structured CNNs called evolutionary exploration of augmenting convolutional topoligies (EXACT) has been presented, which to the author's knowlege

GECCO '17 Companion, Berlin, Germany
 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3076002

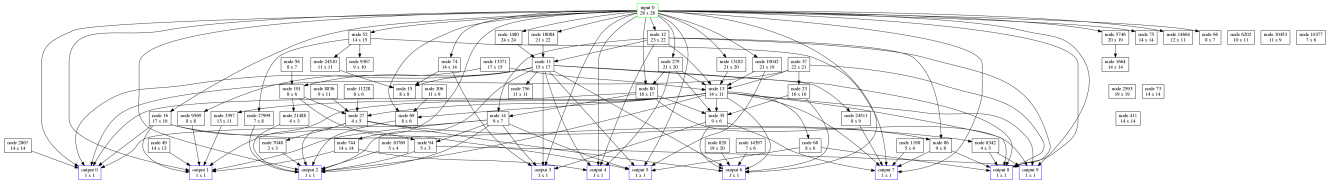

Figure 1: Genome 59455 had the best training accuracy for the epigenetic weight search. This network had a training error of 3,830.97, test error of 633.95, training accuracy of 98.42% and test accuracy of 98.32%.

Network	Number Weights	Training Error			Testing Error		
		Best	Avg	Worst	Best	Avg	Worst
One Layer	7840	16222.59	16637.26	17457.19	2643.16	2686.82	2792.68
Two Layer	8260	7041.03	8063.04	9084.34	1186.11	1331.50	1499.41
Modified LeNet	12285	7994.90	8556.57	9484.93	1325.92	1408.41	1585.14
Network	**Number Weights**	**Training Predictions**			**Testing Predictions**		
		Best	Avg	Worst	Best	Avg	Worst
One Layer	7840	92.02%	91.04%	80.67%	91.97%	91.10%	90.09%
Two Layer	8260	96.66%	96.66%	94.77%	96.66%	95.90%	94.63%
Modified LeNet	12285	97.27%	96.83%	96.09%	97.19%	96.79%	95.90%

Table 1: Benchmark Neural Network Error and Prediction Rates

Network	Avg. Num. Weights	Training Error			Testing Error		
		Best	Avg	Worst	Best	Avg	Worst
Randomized	25,603.35	3,494.54	3,742.22	3,825.23	544.26	603.17	682.75
Epigenetic	23862.65	3,644.30	3,909.88	3,991.73	594.13	657.48	710.25
Network	**Avg. Num. Weights**	**Training Predictions**			**Testing Predictions**		
		Best	Avg	Worst	Best	Avg	Worst
Randomized	25,603.35	97.75%	97.33%	97.05%	97.89%	97.40%	96.98%
Epigenetic	23862.65	98.42%	97.98%	97.48%	98.32%	97.87%	97.28%

Table 2: Error and Prediction Rates for the top 20 CNNs in each EXACT search.

is the first capable of performing this task. In order to overcome the computational demands of evolving large numbers of CNNs, it was implemented as part of the Citizen Science Grid, a BOINC volunteer computing project. Over 4,500 volunteered compute hosts were used to train over 120,000 evolved CNNs during a period of 2 months, which resulted in CNNs reaching 98.32% test accuracy on the MNIST handwritten digits data set, and showing significantly improved training and testing error over human designed benchmark nerual networks. The evolved neural networks show significant differences from the highly structued human designed CNNs found in the literature, containing interesting structures which bear some similarities to biological neurons. The author hopes that these may provide some new insights to the machine learning community in the development of new CNN architectures.

This work opens the door for significant future work. In particular, the EXACT algorithm does not yet evolve pooling layers. This can be done by having each filter perform a pooling operation of an arbrarity size, which can be mutated by additional mutation operations. The algorithm also currently only supports 2 dimensional input and filters, and will need to be updated to utilize 3 dimensional inputs and filters so that it can evolve CNNs for color data sets such as the CIFAR and TinyImage datasets [3, 11].

The use of epigenetic weight initialization, where child CNNs reuse trained weights from their parents has shown potential for improving the CNNs evolved by EXACT, however it did not seem to reduce the number of epochs required by backpropagation to find a minimal training error. This may be because different CNN training hyperparameters may provide more effective for these CNNs. It may also be possible to evolve the hyperparameters used to train the CNNs along with their structure for improved results as done by other recent work [6, 7]. Pretraining the CNNs using restricted boltzmann machines [2] has the potential to even further improve accuracy of the trained CNNs, and may be potentially combined with epigenetic pretraining. Lastly, EXACT evolves and trains a large number of CNNs in each search, which provides an opportunity to determine how robust various CNN training techniques are and to see if these methods have any effect on the structure of the CNNs involved.

ACKNOWLEDGMENTS

We would like to give a special thanks to all the volunteers on the Citizen Science Grid who generously provided their CPU cycles to this research. This work has been partially supported by the National Science Foundation under Grant Number 1319700. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
[2] G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
[3] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 2009.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
[6] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat. Evolving deep neural networks. *arXiv preprint arXiv:1703.00548*, 2017.
[7] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, Q. Le, and A. Kurakin. Large-scale evolution of image classifiers. *arXiv preprint arXiv:1703.01041*, 2017.
[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[9] K. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.
[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
[11] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970, 2008.