



ME Project: Genetic Algorithms for Reducing the Complexity of Deep Neural Networks

Eoin O'Connell (UCD: 13335561)
Monthly Report 1, October 14, 2017

Table of Contents

1	Work Plan	1
2	Git & Backups	1
3	Datasets and trained networks	1
4	State-of-the-art	1

1 Work Plan

Between now and the next monthly report I will continue learning and investigating about deep neural networks. I am enrolled in a number of on-line courses. The machine learning course by Andrew Ng on Coursera [1] and the Google deep learning course on Udacity [2]. Both of these focus on the theory of neural networks but the Udacity course teaches how to use tensorflow also. I have coded in Python before but there will need be some learning on utilising it specifically for data science and machine learning over the coming month. I am also enrolled in UCDs machine learning course this semester which will focus on a broad number of ML techniques as well as neural networks.

I hope to begin work on encoding neural network parameters for use in the GA before the next report.

Beyond the next report I hope to have a fully worked method of encoding the network parameters before returning from the Christmas break. I think this is reasonable as much of the time between now and then will be spent undertaking a crash course on deep neural networks and tackling one of the more complex problems of the project. From then on I hope to begin work on a basic method of neuroevolution whilst simultaneously reflecting on and deciding what novelties I could develop for my algorithm. The stretch from the mid-term break till May will be focussed on implementing that novelty into a fully functional genetic algorithm implementation.

2 Git & Backups

At present any code I am writing is strictly practise code for getting up to speed on the relevant tools and packages for the project. Once I begin development I will initialise a GitHub repository to manage the code.

Simultaneously I utilise cloud backups across all the PCs I use to code and work on the project.

3 Datasets and trained networks

Some datasets with their trained models and state-of-the-art results.

- 96.53% on CIFAR-10 with fractional max-pooling. [3]
- 75.72% on CIFAR-100 using exponential linear units. [4]
- 99.79% on NMIST using DropConnect. [5]

4 State-of-the-art

Neuroevolution implements the automatic design of neural networks for classification or regression using genetic algorithms(GAs) [6]. GAs are a search space method of optimisation following the principals of natural selection. A population of chromosomes are generated, their fitness evaluated and the "fittest" chromosomes are combined to create better offspring for which the process once

again repeats. This algorithm can be successfully applied to the automatic design of neural networks without the need to provide a predefined network topology. These networks are referred to as Topology and Weight Evolving Artificial Neural Networks (TWEANNs).

The majority of state-of-the-art developments in the area of neuroevolution revolve around the improvement of previously developed procedures for automatically developing topologies for Deep Neural Networks (DNNs). The most prominent of these procedures is NeuroEvolution of Augmenting Topologies (NEAT). It proposed a combination of novelties to culminate in an efficient method for evolving topologies along with weights [7]. These novelties remain in more recent approaches such as in CoDeepNeat proposed by Miikkulainen et al [8]. Miikkulainen proposes a method for evolving hyper-parameters (learning rate, training algorithm and data preprocessing) as well as topology and network weights. This new procedure, although computationally intense to execute produces networks comparable to State-of-the-art for image classification on the Cifar-10 data set. NEAT develops a network by first starting minimally and then gradually complexifying the network over generations. This makes for significantly more efficient networks.

Many of NEAT's novelties centred around its use of historical markers in each chromosome to aid in crossover and mutation. Previous GAs faced the Competing Conventions Problem. When chromosomes don't have identical encoding, crossover may cause some loss in functionality. [9] Historical markers allow the algorithm to track identical structures (Extending our biological metaphor this would be similar to tracking species.) and avoid gene duplication or loss of features. The markers can also be used to protect innovation. The introduction of a new connection to a network, through mutation, may negatively impact the fitness of the chromosome but given time to train the connection may improve the fitness overall. Organising chromosomes into species we can apply a method known as explicit feature sharing so that chromosomes with innovations are protected by the species at large and don't have to compete with the global population.

Other offshoot of NEAT include Feature selection and Feature De-selection NEAT (FS-NEAT [10], FD-NEAT [11]). These both focus on similar methods for automating the selection of features as well as network weights and topologies. These methods provide computational efficiency by removing potentially irrelevant or unhelpful features.

In order to reduce the complexity of produced neural networks, it will be important to examine fully connected versus sparsely connected network topologies. Research conducted by Mocanu et al. [12] sets out a different procedure referred to as Sparse Evolutionary Training (SET). This is a specific method for training sparsely connected networks with much smaller computational intensity. The procedure was tested on two common network types, Restricted Boltzman machine and Multi-layer Perceptron, against 14 dataset benchmarks. In each case SET quadratically reduced the complexity of the neural networks whilst maintaining or surpassing the accuracy of the state-of-the-art fully connected networks.

A successful fully connected model referred to as Evolutionary DEep Networks or EDEN has been tested against a number of MNIST offshoot datasets and equalled or in some cases surpassed the state-of-the-art [13]. This approach is significantly different than previous approaches as it reduces the complexity of the genetic algorithm itself to run on a single powerful GPU. The algorithm will require a method of evaluating the fitness of the generated networks. EDEN utilises a tournament style fitness function where only a small portion of the population each generation is evaluated the best are used to generate offspring.

To develop a method of neuroevolution we will need to encode the parameters of a DNN into chromosomes to apply the methods of natural selection (Speciation, mutation, crossover etc.). Both NEAT and CoDeepNEAT encoded the genes using a graph structure. Where they differed

however was that in CoDeepNEAT each gene represented a layer as oppose to a single neuron like with NEAT.

References

- [1] Ng. (2016) Stanford course on machine learning. [Online]. Available: <https://www.coursera.org/learn/machine-learning>
- [2] Vanhoucke. (2017) Google course on ml and tensorflow. [Online]. Available: <https://classroom.udacity.com/courses/ud730>
- [3] B. Graham, "Fractional max-pooling," *CoRR*, vol. abs/1412.6071, 2014. [Online]. Available: <http://arxiv.org/abs/1412.6071>
- [4] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015. [Online]. Available: <http://arxiv.org/abs/1511.07289>
- [5] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, S. Dasgupta and D. Mcallester, Eds., vol. 28, no. 3. JMLR Workshop and Conference Proceedings, May 2013, pp. 1058–1066. [Online]. Available: <http://jmlr.org/proceedings/papers/v28/wan13.pdf>
- [6] K. O. Stanley. (2017) Neuroevolution: A different kind of deep learning. [Online]. Available: <https://www.oreilly.com/ideas/neuroevolution-a-different-kind-of-deep-learning>
- [7] K. O. Stanley and R. Miikkulainen, "Efficient reinforcement learning through evolving neural network topologies," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO'02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 569–577. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2955491.2955578>
- [8] R. Miikkulainen, J. Z. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat, "Evolving deep neural networks," *CoRR*, vol. abs/1703.00548, 2017. [Online]. Available: <http://arxiv.org/abs/1703.00548>
- [9] N. J. Radcliffe, "Genetic set recombination and its application to neural network topology optimisation," *Neural Computing & Applications*, vol. 1, no. 1, pp. 67–90, Mar 1993. [Online]. Available: <https://doi.org/10.1007/BF01411376>
- [10] S. Whiteson, P. Stone, K. Stanley, R. Miikkulainen, and N. Kohl, "Automatic feature selection in neuroevolution," 2005, pp. 1225–1232, cited By 46. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-32444438066&doi=10.1145%2F1068009.1068210&partnerID=40&md5=bf49e8b65e4fe639405812a597dee578>
- [11] M. Tan, R. Deklerck, B. Jansen, M. Bister, and J. Cornelis, "A novel computer-aided lung nodule detection system for ct images," *Medical Physics*, vol. 38, no. 10, pp. 5630–5645, 2011. [Online]. Available: <http://dx.doi.org/10.1118/1.3633941>

- [12] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, "Evolutionary training of sparse artificial neural networks: A network science perspective," *CoRR*, vol. abs/1707.04780, 2017. [Online]. Available: <http://arxiv.org/abs/1707.04780>
- [13] E. Dufourq and B. A. Bassett, "EDEN: Evolutionary Deep Networks for Efficient Machine Learning," *ArXiv e-prints*, Sep. 2017.