A GPU based distributed algorithm for HyperNEAT

Emad Hosseini * Ali Kamandi

June 29, 2019

1 Introduction

The advancements in the field AI in the past few decades has led to it being one of the tools of our everyday life. Yet there are some great challenges in developing and training AI systems. There are some tasks that today's AI systems are particularly more capable of doing. That includes image classification [1], natural language processing [2], motor control [3] and many other fields that previously seemed impossible for machines to do and were considered specific to humans and animals.

But what made these traits possible is the advancements in computers and hardware that made faster processors and larger memories and with the help of more data scientists could build working AI softwares.

Having more data and enough time to process that data is not something to come by easily and in many cases like autonomous navigation in unknown environments or critical decision making for self driving vehicles lack either the data or the time to train conventional neural network models.

One group of models that tackles these challenges are neuroevolution models. That is defined by Gomez and Miikkulainen in 1999 as "systems that evolve neural networks using genetic algorithms" [4]

In this paper we will introduce a computation method for a specific type of NE model using general purpose computers. In the next part we see what has already been done in this field and after that in section 3 our computational method is presented. Some benchmarks are done and results are shown in section 4 with conclusions that follows.

*University of Tehran, Department of Algorithms and computation.

2 Background

Neuroevolution of augmented topologies (NEAT) is one of the more successful models of neuroevolution introduced in 2002 by Stanley and Miikkulainen [5] in this model there is an encoding of the neural network in a genotype that considers an

A population of such genomes and their respective phenotypes is then created and using genetic algorithm this population is then directs towards better networks that work better to solve the problem at hand.

NEAT is really successful in finding minimal networks for many tasks that are simple enough but when the requirement of the task is more than that, the search space gets big enough that NEAT is inefficient in finding the best networks. This was addressed in another model called HyperNEAT [6] that uses underlying symmetries in tasks

The nature of GA involves many simple individuals controlled by an environment that produces the survival of the fittest mechanism for a certain goal. This seems an ideal task for a distributed system.

References

[1] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[2] Erik Cambria and Bebo White. Jumping nlp curves: A review of natural language processing

research. *IEEE Computational intelligence magazine*, 9(2):48–57, 2014.

[3] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338, 2016.

[4] Faustino J Gomez and Risto Miikkulainen. Solving non-markovian control tasks with neuroevolution. In *IJCAI*, volume 99, pages 1356–1361, 1999.

[5] Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

[6] Kenneth O Stanley, David B D'Ambrosio, and Jason Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial life*, 15(2):185–212, 2009.