# DarwiNN: Efficient Distributed Neuroevolution under Communication Constrains

## Background

Evolution Strategies (ES) are a sub-class of nature-inspired direct search (and optimization) methods which belong to the evolutionary algorithms (EAs) which we covered in week 9. These use mutation, recombination, and selection applied to a population of individuals containing candidate solution to evolve iteratively better solutions. In the case of DarwiNN it has been applied to synaptic strength of connections to decrease the number of nodes in the nueral network by a factor of 40%.

MPI - Message Passing Interface (lower level than most parallel programming libraries). Leveraged by authors to parallelize the MPI component of the chromosome update.

## Motivation and background

Malik, Petrica, Kapre and Blott look at applying neuroevolution (NE) which is defined as the application of evolution-based training methods to Deep Neural Networks. NE is highly parallel as noted by the authors and can therefore be utilized in large scale distributed inference-specific hardware in the cloud.

The authors introduce a chromosome update (CU) which is a communication-optimized method for distributing NE computation and DarwiNN for distributed neuroevolution.

This short paper introduces the distributed algorithm and bi-inspired approaches.

## Research objective/question(s)

Neuro-Evolution (NE) an evolution-based alternative to backpropagation for training the weights of deep neural networks. This approach is noted as being successfully implemente in reinformacenet learning and also image classification problems.

The researchers develop a novel approach to take advantage of distributing NE computation across a cluster of computers.

The authors believe with a NE Chromosome Update that they can achieve optimal scalability.

## Prior relevant work/literature gap

Karel Lenc, Erich Elsen, Tom Schaul, and Karen Simonyan. 2019. Non-

Differentiable Supervised Learning with Evolution Strategies and Hybrid Methods.

CoRR abs/1906.03139 (2019). arXiv:1906.03139 <http://arxiv.org/abs/1906.03139>

* In this work the authors show that evolution strategies (ES) are viable for learning non-differentiable parameters for large, supervised models
* Hybrid approach that used gradient-based methods for differentiable weights and ES for non-differentiable (sparsity masks) to train sparse models from scratch.
* This method uses semi-updates execution model

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017.

Evolution strategies as a scalable alternative to reinforcement learning. arXiv

preprint arXiv:1703.03864 (2017).

* Novel communication strategy based on random numbers
* Only scalers are communicated allowing scale to over a thousand parallel workers
* Found evolution strategies are highly parallelizable – achieving linear speedup
* Match Atari performance with 3x – 10x data volume
* Games trained with 1 billion frames; 1-day is 320 million frames
* Parallelized across 720 CPUs (training to 1-hour) – performance for half were better.

Xingwen Zhang, Jeff Clune, and Kenneth O. Stanley. 2017. On the Relationship

Between the OpenAI Evolution Strategy and Stochastic Gradient Descent. ArXiv

abs/1712.06564 (2017).

* ES can rival performance of SCD (stochastic gradient descent)
* Goal of authors is reduced ambiguity in comparing ES to SCD models
* Leverage supervised problem without domain noise to measure the correlation between gradient computed by ES and SCD
* Demonstration of achieving 99% accuracy using ES

## Theory, conceptual framing

The Chromome Update represents the novel approach to distributing across multiple machines. Each N work process executes the following algorithm.

Text

Description automatically generated

DarwiNN has been updated since the original paper and can distribute the computation of any of its optimizers on an arbitrary number of GPUs, using the following distribution patterns:

* data-parallel inference with sequential evaluation of population individuals, which we call Distributed Data Parallel (DPP). In this distribution mode, parallelization is across the input batch, and is limited by the size of said batch. E.g. when training with batch size 64, the maximum number of GPUs is 64
* population-parallel evaluation, which we call Distributed Population Parallel (DPP), whereby the population is distributed across the available GPUs, and the maximum number of GPUs is equal to the population size.
* Semi-updates DPP, described here, which more effectively distributes the gradient estimation step of ES
* Chromosome-updates DPP, which is similar to Semi-updates but minimizes communication requirements and scales better under network bandwidth constraints

## Methods

Diagram

Description automatically generated

* DarwiNN is GPU-accelerated neuroevolution library designed to work with supervised learning flows based on PyTorch, the ENV procded distributd communication over MPI collective
* Chromosome Update requires 2x less communication compared to the semi-update method employed previously.

## Results

The researchers leveraging previous work (MNIST – 3M parameters, CIF 900K parameters, and CIF 8M parameters) to test the efficiently of chromosome update versus the semi-update method. The test was run on a distributed platform of 48 notes (or 96 GPUs) and a fabric network. The goal is to see the efficiency with a scaling (e.g. network bandwidth). The chromosome update performs better when there are network constraints.

The problem encountered is the generation of the noise matrix becomes a bottleneck as the number of workers increase (see algorithm line 1).

Chart, line chart

Description automatically generated

## Discussion

The big problem the authors were trying to solve in semi-update is the significant increase in inter-process communication requirements. By cutting the communication by 2x the algorithm is noticeably more efficient especially under network constrained conditions.

## Thoughts

The authors were influenced on communication approach and optimizing that part of the process to enable parallelization as noted from the prior work in ES. To make it viable they need to deal with a practical solution to the noise matrix (computational complexity and memory). It generates gaussian noise matrix for each instance in the population