Neural Network Topology and Weight Optimization through Neuro Differential Evolution

Karl Mason*
National University of Ireland Galway
Discipline of Information Technology
Galway, Ireland
k.mason2@nuigalway.ie

Jim Duggan
National University of Ireland Galway
Discipline of Information Technology
Galway, Ireland
jim.duggan@nuigalway.ie

Enda Howley
National University of Ireland Galway
Discipline of Information Technology
Galway, Ireland
ehowley@nuigalway.ie

ABSTRACT

This research proposes a novel algorithm, Neuro Differential Evolution (NDE), to optimize the topology and weights of neural networks. NDE makes a clear distinction between neural network topology optimization and weight optimization. A genetic algorithm (GA) is implemented to optimize the network topology as this is a discrete problem while differential evolution (DE) is applied to the network weights, which are continuous variables.The results presented in this paper demonstrate that this combined approach can successfully grow neural networks, from just a single neuron, that can produce feasible solutions when other methods fail. NDE outperforms the current state of the art neuroevolution algorithms on a range of increasingly complex reinforcement learning problems.

CCS CONCEPTS

•Mathematics of computing → Evolutionary algorithms; Bio-inspired optimization; **•Computing methodologies → Neural networks;** *Artificial intelligence; Genetic algorithms;*

KEYWORDS

Genetic Algorithms, Differential Evolution, Neuroevolution

ACM Reference format:
Karl Mason, Jim Duggan, and Enda Howley. 2017. Neural Network Topology and Weight Optimization through Neuro Differential Evolution. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3075967

1 INTRODUCTION

Neuroevolution (NE) is a branch of computer science that applies evolutionary methods to neural networks [1]. This is useful for a wide range of problems, from classification to control [4, 5]. Neuroevolution methods have been shown to be particularly effective at solving reinforcement learning tasks. Multiple studies have been conducted, revealing that neuroevolution algorithms outperform several state of the art reinforcement learning algorithms on problems such as the pole balancing problem [2].

Two of the best performing neuroevolution algorithms include NEAT (NeuroEvolution of Augmenting Topologies) [5] and ESP (Enforced SubPopulations) [3]. The NEAT algorithm uses a GA to evolve both the network topology and weights. It employs a direct encoding scheme and begins with a minimal network with no hidden neurons. Complexity is added to the network by adding neurons and connections via mutation. Genes are also divided into species. ESP uses a fixed neural network size. In ESP, the population is divided into distinct sub populations. Networks are formed by selecting a neuron from each sub population and combining them to form a network. Crossover only occurs between chromosomes within the same subpopulation. No inter sup population breeding is allowed. Offspring also remain within the parents' sub population.

Many neuroevolution methods, including NEAT, use a genetic algorithm (GA) to optimize both the topology and weights of the network. It is hypothesised in this paper that this is not the most sensible approach. This paper will propose the novel NDE algorithm (Neuro Differential Evolution) that optimizes the network topology using a GA and optimizes the network weights using Differential Evolution. The contributions of this paper are : 1) The application of differential evolution to topology and weight evolving artificial neural networks. 2) Combining genetic algorithms with differential evolution in a meaningful way that exploits their respective strengths, to evolve neural networks. 3) To establish what the limitations of current neuroevolution algorithms are for solving the pole balancing problem, i.e. at what point do these algorithms fail to find acceptable solutions.

2 NEURO DIFFERENTIAL EVOLUTION (NDE)

The key insight of NDE is to address the tasks of topology and weight optimization separately. Selecting the correct neural network topology is a discrete optimization problem while choosing the correct weights for the network is a continuous problem. NDE uses a genetic algorithm to optimize the topology of the network and differential evolution to optimize the weights of any given network topology. The second advantage of NDE is that a network size does not need to be selected. NDE effectively grows a neural network by beginning with a network of just one neuron. New species with extra neurons are added at each iteration with a certain probability or if the progress of the algorithm stagnates. This ensure that the smallest suitable network is found. The pseudocode in Algorithm 1 describes how the algorithm functions.

*Corresponding Author

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3075967

```
Create c chromosomes for species of 1 neuron
while Eval e <maxEvals & fitness < threshold do
  while numberOfSpecies > maxSpecies do
    Remove worst species
  end
  if lastImprovement > stagnationTime then
      Create new species with extra neuron
      lastImprovement = 0
  end if
  lastImprovement++
  for Species = 1 to S do
    for Chromosome c = 1 to C do
      Apply DE to weights of c
      if Fitness > bestOverallFit then
          bestOverallChromosome = c
          lastImprovement = 0
      end if
      if Fitness > bestSpeciesFit then
          bestSpeciesChromosome = c
      end if
    end
    ADD bestSpeciesChromosome to next gen
    while nextGenSize < currentGenSize do
      SELECT parents for mating
      Perform crossover
      MUTATE child chromosome with prob, mp
      ADD child chromosome to next gen
    end
    if rand() < sm then
        Create new species with extra neuron
    end if
  end
end
Return best Network
```

Algorithm 1: Neuro Differential Evolution (NDE)

3 EXPERIMENTS

The performance of the proposed NDE will be compared to NEAT and ESP on four variants of the non markovian double pole balancing problem. Each algorithm must balance two poles for a predetermined amount of time with no velocity information, only position information relating to the cart and the pole. At each variation of the problem, the short pole length will be incrementally increased. This will increase the difficulty of the problem. If the system is balanced for 100,000 time steps (30 minutes of simulated time), the controller must then pass a generalization test. The purpose of this test is to determine if the system can recover from a variety of different starting positions. The generalization test will examine 625 distinct initial states. The evolved controller must be able to balance the system for 1,000 time steps in at least 200 of the 625 initial states. The pole balancing experiment implemented in this research is the same as that used in previous experiments [3, 5]. All 4 configurations of the non markovian double pole balancing problem were evaluated over 10 runs, each run consisting of 10^6 evaluations of the pole balancing problem.

4 RESULTS & CONCLUSION

Table 1 shows that for the simplest variant of the pole balancing problem, NEAT is the best performing algorithm. As the short pole increases, NDE significantly outperforms ESP and NEAT when compared using the t-test with a significance threshold of 5%. This is also reflected in the number of failed runs presented in Figure 1.

Table 1: Average Evaluations Taken for 4 Variants of Non Markovian Double Pole Balancing Problem.

Pole Increase	NDE Avg (StDev)	NEAT Avg (StDev)	ESP Avg (StDev)
0	46313.8 (32991.3)	30945.3 (22574.8)	64905.5 (38867.8)
1	99028.5 (109294.0)	239714.6 (401160.3)	150851.1 (57237.8)
2	167100.3 (111380.5)	459959.4 (447950.8)	327049.6 (260866.7)
3	279708.6 (232386.9)	918183.3 (258727.1)	453825.4 (169078.3)

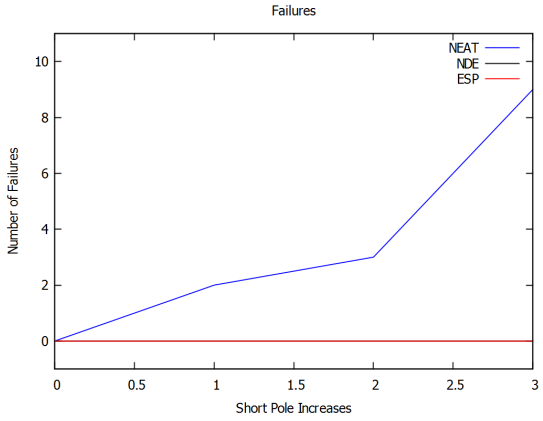

Figure 1: Short Pole Length Increases vs Number of Failed Runs.

In summary, the contributions of this research are: 1) Differential evolution can be applied to topology and weight evolving artificial neural networks. The proposed NDE performs better that state of the art methods for very difficult reinforcement learning tasks. 2) Separating the optimization of network topologies and weights into two distinct but connected optimization problems can yield superior results. This paper presents a novel hybrid algorithm for neural network optimization using genetic algorithms and differential evolution. 3) NDE scales better to more difficult problems than NEAT and ESP.

REFERENCES

[1] Dario Floreano, Peter Dürr, and Claudio Mattiussi. 2008. Neuroevolution: from architectures to learning. *Evolutionary Intelligence* 1, 1 (2008), 47–62.

[2] Faustino Gomez, Jürgen Schmidhuber, and Risto Miikkulainen. 2008. Accelerated neural evolution through cooperatively coevolved synapses. *Journal of Machine Learning Research* 9, May (2008), 937–965.

[3] Faustino J Gomez and Risto Miikkulainen. 1999. Solving non-Markovian control tasks with neuroevolution. In *IJCAI*, Vol. 99. 1356–1361.

[4] César Hervás, Francisco J Martínez, and Pedro Antonio Gutiérrez. 2006. Classification by means of evolutionary product-unit neural networks. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 1525–1532.

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