Evolving Deep Neural Networks : A New Prospect

Sreenivas Sremath Tirumala, Ali S and C Phani Ramesh

Christophe Thao Ky
Student ID: 17R50513

Introduction

- DNN training is time consuming
- This issue was resolved for ANN using Evolutionary Computation (EC)
- Why not apply the same to DNN?
- Specifically to topology and weight initialization

Overview

1. Introduction
2. Genetic algorithms
3. Two co-evolutions strategies studied
4. The proposed solution
5. Improvements of the implementation
6. Experiments
7. Evaluation
8. Conclusion

Genetic algorithms

- Inspired by evolution -> natural selection
- Survival of the fittest
- Population-based trial and error

Genetic algorithms

Algorithm :

1. Initialize a set of solutions
2. Does it satisfy the constraint ? If yes then stop
3. Pick the best(s) and discard the rest
4. Generate a new set of solutions based on the bests with **small** random changes
5. Go to step 2

Genetic algorithms

Examples :

- GeNeralized Acquisition of Re-current Link (GNARL)
- Cellular Encoding
- EPNet
- Neuro Evolution Augmenting topologies (NEAT)
- Real time NEAT (rtNEAT)
- HyperNEAT
- Content-Generating NEAT(cgNEAT)

Two co-evolution strategies

Competitive co-evolution:

- Individuals compete among each other in a population
- Race to the top position
- Limit : relative fitness

Two co-evolution strategies

Cooperative co-evolution:

- Divide problem into subproblems
- Optimize subsolutions of said subproblems
- Recombine subsolutions

Two co-evolution strategies

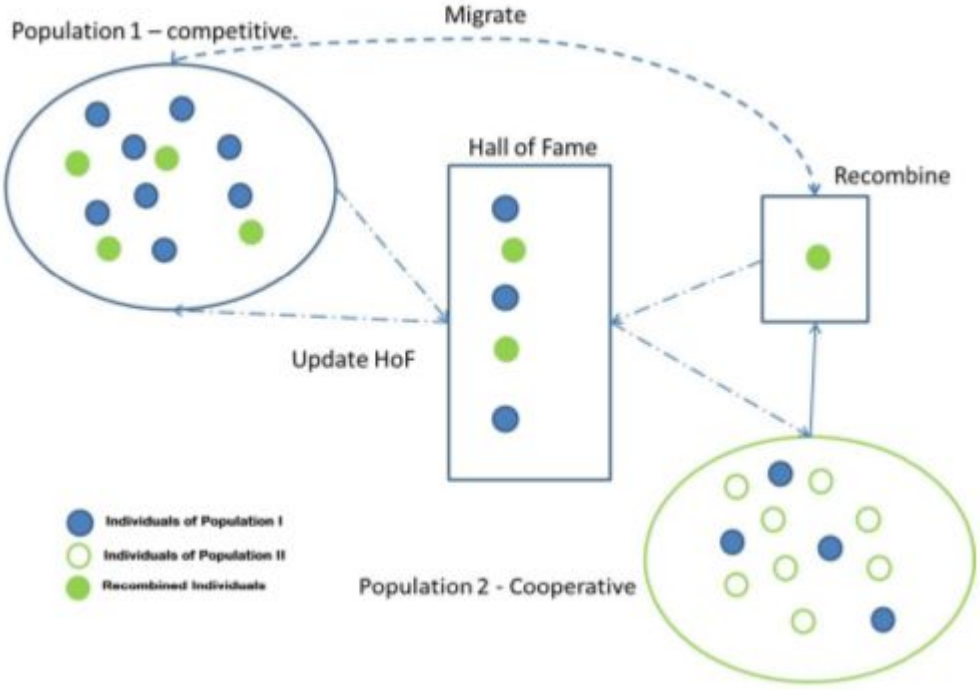
Issue:

- Competing conventions problem:

$$[A,B,C] + [C,B,A] = [C,B,C] \text{ or } [A,B,A] \text{ or…}$$

- For **n** hidden nodes, **n!** functionally equivalent solutions

The proposed solution



Improvements of the implementation

- Number to trace back the origin of the solution to avoid remigrating to native population
- Attribute to encourage combination with solution from another population
- Hall of fame: keep the best solutions from previous generations

Experiments

- Did not implement said neural network…

- 3 5-layered Deep Neural Network with 784 nodes each:
  - DNN-R : random weights
  - DNN-CCEA : competitive co-evolution algorithm
  - DNN-COCA : cooperative co-evolution algorithm

- Dataset used : MNIST, IRIS

Evaluation

Strategy	Accuracy (%)		Error Rate (%)	
	MNIST	IRIS	MNIST	IRIS
DNN-R	94.3	96.9	0.54	0.31
DNN-CCEA	96.3	98.1	0.201	0.19
DNN-COCA	98.7	98.3	0.12	0.131

Evaluation

Strategy	Avg. Time (hours)	Avg. Performance
DNN-R	14.5	2.5
DNN-CCEA	8.3	1.19
DNN-COCA	10.2	1.73

Training time and performance for MNIST

Conclusion

- The authors have proposed a method yet to be experimented
- The potential of Evolutionary Computation ?
- It certainly makes learning procedure faster
- Need more study