

Introduction to Neuro-Evolution: Solutions

Andrew Geng, Jonas Klare, John Balis

December 13, 2018

1 Simulated Annealing

1.1

Possible observations:

1. No, many of the points ended being stuck within the local minimums with only a small portion of the points ending up in the actual global minimum.
2. Points got stuck within the local minimum due to the gradient descent algorithms converging to the nearest minimum and not being able to jump out of the given local minimum.

1.2

Possible observations:

1. Yes, more points reached the global minimum in contrast to the gradient descent algorithms.
2. This is due to the mutation like property in which simulated annealing probabilistically tests other points which might be none optimal immediately .
3. Simulated annealing algorithms effectively create possibilities of jumping out of local minimums.

1.3

1. A traditional gradient descent algorithm will find the immediate optimal minimum in any given situation. However, this minimum is not guaranteed to be the global minimum as gradient descent can easily get stuck in local minimums. However due to the randomized nature of mutations and simulated annealing, we are able to jump out of the local minimum and test other possible minimums.

2 Mutation and Speciation

2.1

The children are part of the parents, the first part of the a child is one parent, and the next two are the other. A reasonable child would be [1,.1,5] or [2,.5,3]

2.2

These ones have one value that are now a random value compared to the other children, even though they are from the same parents

2.3

Fitness function looks to accurately model some part of the real problem, which is a gradient descent solver over things with multiple minima. Children on average perform much better than the parents.

2.4

Genetic algorithm outperforms initial parents. It is an effective way to get out of local minima, however it takes a lot of computing power to create these agents, especially as population size increases. PROS: avoids local minima well CONS: computing time/power.

3 Neuro-Evolution

3.1

Possible observations:

Either none of the training runs results in significant survival time or one or more training run results in significant survival time, and can be seen to exhibit intelligent some sort of behavior with the intent of avoiding the hunter.

It's important to note that neuro-evolution is very unpredictable, and may fail to find a good solution even after a very long time. However, running a given training run for longer can only improve the performance of the network.

3.2

You should observe that the sample network has a very effective a deliberate style of avoiding the hunter. All successful SSNs we have trained have employed a variant of this style. It's important to note that while is no upper bound on how long it takes to train a network to this quality and we could not leave it's generation as an excercise, it is possible to train a network to this quality in even 1000 or fewer epochs.

3.3

You will likely observe that the best score is worse for the agents trained from scratch than when training with a hunter speed of .2. The pre-trained agent should perform similarly with both speeds. The takeaway here is that sometimes it is useful to initially reduce the difficult of training simulations and gradually increase difficulty as performance increases.