Review of: Efficient Reinforcement Learning through Evolving Neural Network Topologies

Alexander Sverdlov <alex@theparticle.com>

May 11th, 2004

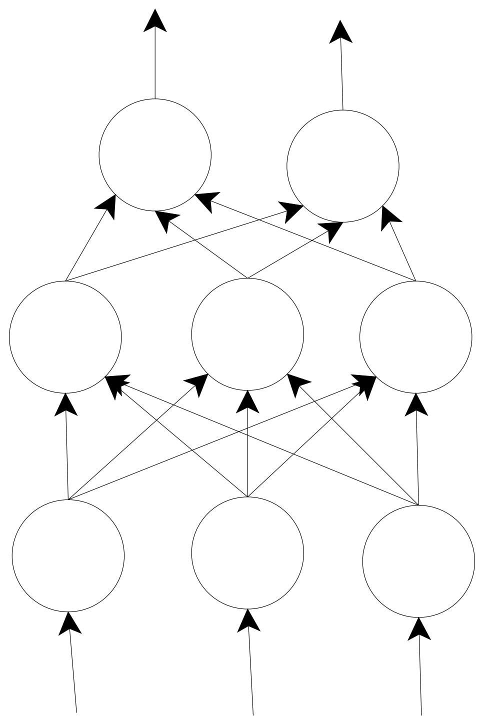
Abstract

This talk reviews a paper by Kenneth O. Stanley, and Risto Miikkulainen, titled "Efficient Reinforcement Learning through Evolving Neural Network Topologies."

Introduction

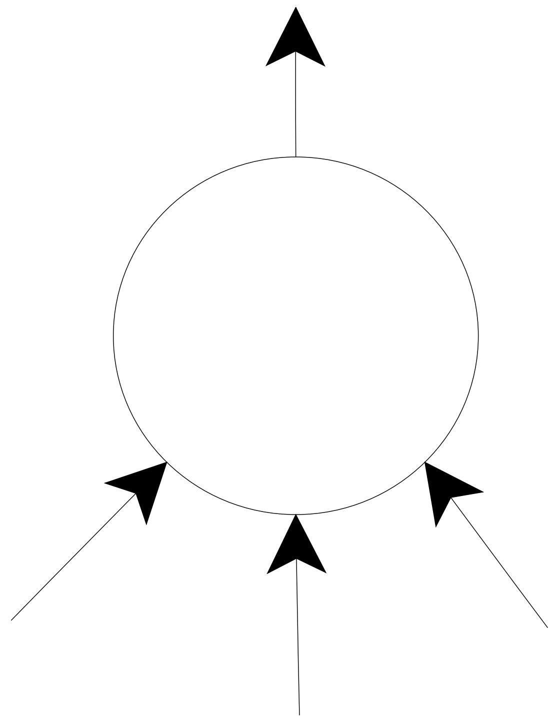
- Neural Networks
- Neuroevolution (NE)
- Neuroevolution of Augmenting Topologies (NEAT)

Neural Networks



- A network of nodes, with weighted connections.
- Some nodes are designated as “input” or “output”.
- Nodes that are neither input nor output are “internal” nodes.

Neural Networks Cont.



- Each node is the same as every other node.
- Functionality is determined by weights on connections between nodes.
- Neural Networks are universal function approximators.
- Need to be trained. Traditional approach: Backpropagation.
- No good way of finding global maxima.

Evolutionary Approach to “Training”

- Concept of Genotype and Phenotype.
- Fitness function.
- Randomization.
- Populations.

NeuroEvolution

Artificial evolution of neural networks using genetic algorithms.

Start with a ‘random’ population (many networks with random weights).

1. Randomly cross breed their genomes, ie: weights.

2. Introduce mutations.

3. Apply fitness function to eliminate weak breeds.

4. Repeat.

NeuroEvolution Cont.

- Only weights are changed.
- There is an assumption that we chose the right network topology.
- We need to start with a sufficiently large network to solve non-trivial problems.
- Large populations of large networks are hard to converge.
- Number of hidden nodes plays a huge role in whether the network converges.

NeuroEvolution of Augmenting Topologies

A method designed to take advantage of structure as a way of minimizing the dimensionality of the search space of connection weights.

- Also have network structure (as well as weights) as part of the genome.
- Breeding and mutations effect weights as well as network structure.

Issues

1. Is there a genetic representation that allows disparate topologies to crossover in a meaningful way?

2. How can topological innovation that needs a few generations to optimize be protected so that it does not disappear from the population prematurely?

3. How can topologies be minimized *throughout evolution* without the need for a specially contrived fitness function that measures complexity?

Genotype to Phenotype Mapping

Two types of Genes: Node Genes, and Connect Genes.

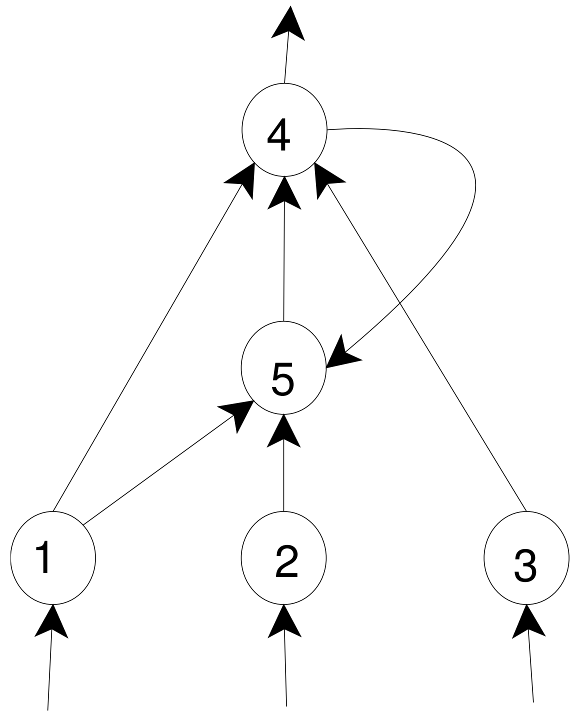
Node Genes:

Node 1 Sensor	Node 2 Sensor	Node 3 Sensor	Node 4 Output	Node 5 Hidden

Connection Genes:

In 1 Out 4 Wght .7 Enabled Innov 1	In 2 Out 4 Wght .5 Disabled Innov 2	In 3 Out 4 Wght .5 Enabled Innov 3	In 2 Out 5 Wght .2 Enabled Innov 4	In 5 Out 4 Wght .4 Enabled Innov 5	In 1 Out 5 Wght .6 Enabled Innov 6	In 4 Out 5 Wght .6 Enabled Innov 11

Genotype to Phenotype Mapping Cont.



- Representation allows for weight as well as structural evolution & mutation.

Structural Mutation

- Mutation may add a connection where none existed before.
  - Grow the connection genome, by adding a connection with a random weight.
- Mutation may add a node.
  - Find a connection to split.
  - Replace that connection with a node connected by two new connections.

Crossover: Tracking Innovations

For crossover, we can match up two genes by their innovation numbers.

- Line up two genes such that their innovation numbers line up.
- The offspring is a gene formed by merging nodes (with the same innovation number) from a more fit parent.
- Excess and disjoint nodes are added from the more fit parent.

This enables meaningful neural network crossover.

Protecting Innovation Through Specialization

- Divide population into species.
- Have individuals compete within their species.
  - Enables structures to optimize to a particular niche.

"The number of excess and disjoint genes between a pair of genomes is a natural measure of their compatibility. The more disjoint two genomes are, the less evolutionary history they share, and thus they less compatible they are."

Minimizing Dimensionality

Instead of starting with a large NN and then trying to adjust weights, NEAT starts with a small genome, and via crossover and mutations grows the network as well as adjusts weights.

Pole Balancing

- Pole Balancing is a *real* task.
- Difficulty can be adjusted.

Double Pole Balancing

- Balancing a single pole is too easy.

Problem: A cart with 2 poles, has a position and velocity. Each pole has an angle. Control is possible since poles have different heights and respond differently to control inputs.

Double Pole Balancing Cont.

Two versions of the 'test':

1. With velocity inputs.
2. Without velocity information.

The one with velocity inputs is Markovian and many different systems have been able to solve it.

Without velocity, the problem is considerably harder; the network must estimate system state in lieu of velocity, which requires recurrent connections.

Double Balancing with Velocities

Many other systems solve this problem. Time-wise, NEAT isn't significantly better, but it managed to evolve a smaller NN (0 to 4 hidden nodes) as opposed to the 2nd runner up (ESP) which evolved 10 hidden nodes.

Double Balancing without Velocities

Only 2 other systems can solve this: CE (Cellular Encoding) and ESP (Enforced Subpopulations)

- Needs special fitness function—can solve the problem by moving cart back and forth quickly, which doesn't require computing velocities.

NEAT takes 25 times fewer evaluations than CE, and 5 times faster than ESP. This shows that evolving topologies has a significant impact on performance.

Future Work

Among other things, NEAT handles complexification relatively well. Meaning that it can incrementally gain new functionalities without loosing what it has learned.

Other directions include applying the same genome (innovation coding) idea to other things besides Neural Networks.

Conclusion

“The main conclusion is that evolving structure and connection weights in the style of NEAT leads to significant performance gains in reinforcement learning.”

The End