Neuroevolution

War of the Artificial Minds

Disclaimer

- I will not be going into depth on Artificial Neural Networks
- I will not be going into depth on Genetic/Evolutionary Algorithms
- Note that I will switch between algorithm and network when talking about Neuroevolution. In general
  - Algorithm will refer to the system as a whole
  - Network will refer to the ANNs generated by the system

What are Neuroevolutionary Algorithms

- ANNs + Genetic Algorithms = Neuroevolutionary Algorithms
- Based of the natural evolution of the nervous system
- Neuroevolutionary Algorithms make use of Genetic Algorithms to build ANNs

Why Should I Care About Neuroevolutionary Algorithms

- They can and probably will drive you car one day
  - https://www.youtube.com/watch?v=5lJuEW-5vr8
  - https://www.youtube.com/watch?v=p_H2TLG1cMo
  - https://www.youtube.com/watch?v=n27Wz9J_WL4

Why Should I Care About Neuroevolutionary Algorithms

- Can work Unsupervised or use Reinforcement Learning
  - Supervised Learning can still be used
  - Optimal actions do not have to be known to train
- Neuroevolutionary Networks have proven to be faster, more efficient, and able to generalize better than other AI methods
- Neuroevolutionary Networks are much more tolerant noise the traditional ANNs
- Neuroevolutionary Networks can continue learning through environmental input well past their initial training

Why Should I Care About Neuroevolutionary Algorithms

- Can be used to combine Expert Networks to solve problems
    - Like the Geth from Mass Effect
- Can be used to design effective ANN topologies
- Due to its imitation of biological nervous systems, Neuroevolutionary Algorithms can be used to research how intelligence formed

Where are Neuroevolutionary Algorithms Used

- Robotics
- Artificial Life
- Games
- Biology Research
  - Can be used to simulate the formation of intelligence

How do I Make Neuroevolutionary Algorithms

- Neuroevolutionary Algorithms, as a concept, is fairly simple
- Apply a Generic Genetic Algorithm to a Population of Randomly generated ANNs
- Choosing how to Encode the Genomes is the Hard Part
- Small Little Tricks can be Used to Make Neuroevolutionary Algorithms Even Stronger

Encoding

- Encoding Comes in Two Flavors
    - Direct Encoding
    - Indirect Encoding

Direct Encoding

- Simplest of the Two Methods
- Also Comes in Two Flavors
  - Conventional Neuroevolution (CNE)
  - Topology & Weight Evolving Artificial Neural Network (TWEANNs)

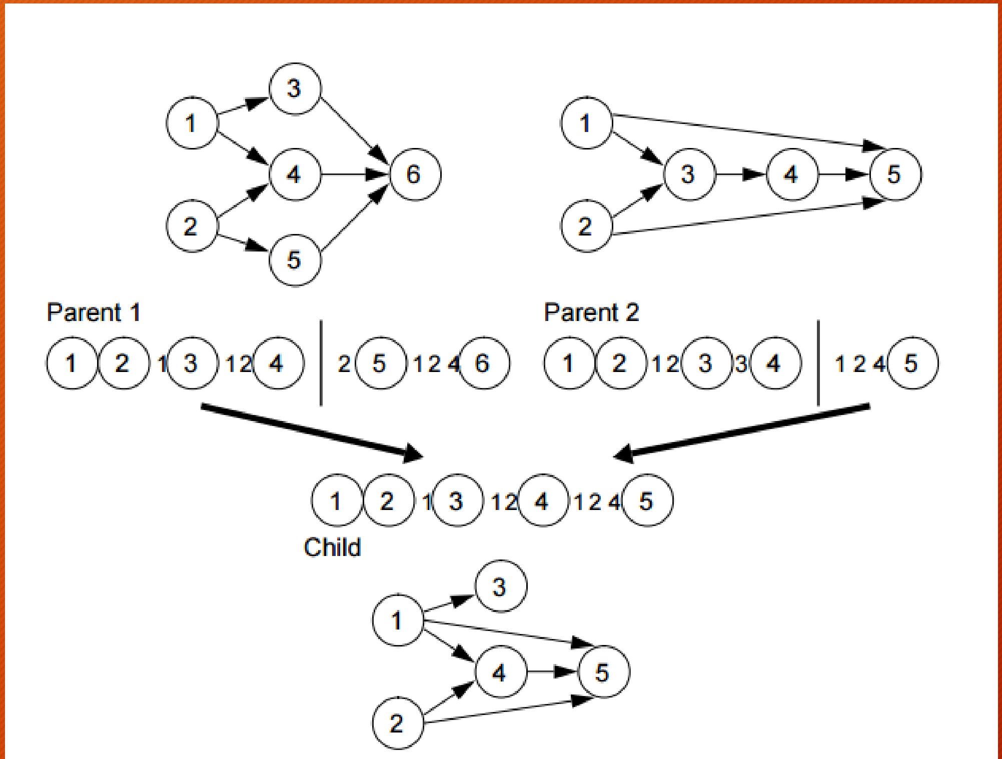
Direct Encoding : (CNE)

- A Fixed ANN Topology is Used on All ANNs
- Edge Weights make up the Genome
- Advantages
  - Simplest to Implement
- Disadvantage
  - Scale Poorly
  - Can get Stuck at Local Optima
  - Dependent on the Chosen ANN Topology
    - ANN Topology can make a large difference in if a problem can be solved or not, but there are no good rules of thumbs for building them

Direct Encoding : TWEANN

- Evolves the ANN Topology as Well as Weights
- Advantages
  - This method makes extremely effective ANN Topologies for the problem it is solving better than any person can
  - Scales really well
  - Unlikely to get Stuck at Local Optima
- Disadvantages
  - More Complex Genome

Direct Encoding : TWEANN



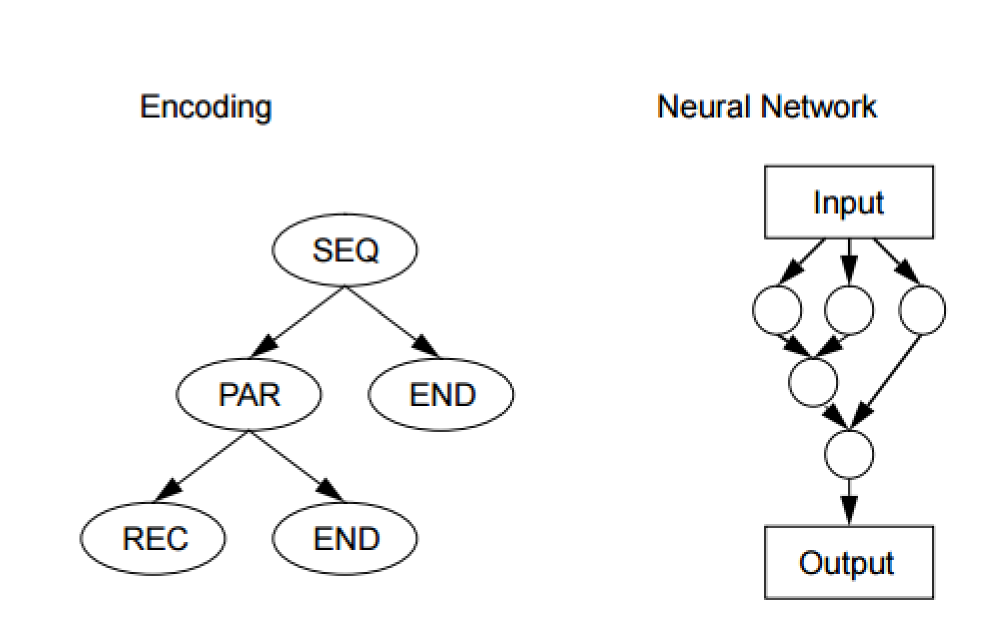
Indirect Encoding

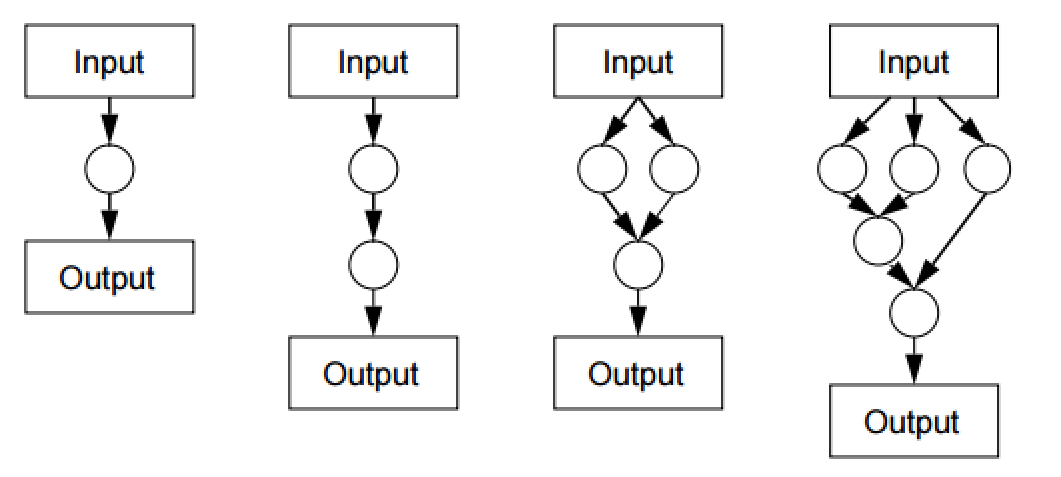
- Genomes that Encode how to Build the ANNs Instead of Data About the ANNs
- Models Natural Nervous Systems Closer than Direct Encoding Methods
- Advantages
  - Can be Highly Compact
  - Can Take Advantage of Modular Solutions
  - Can, Theoretically, Create ANNs the Size of the Human Brain
- Disadvantage
  - Extremely Complex

Indirect Encoding

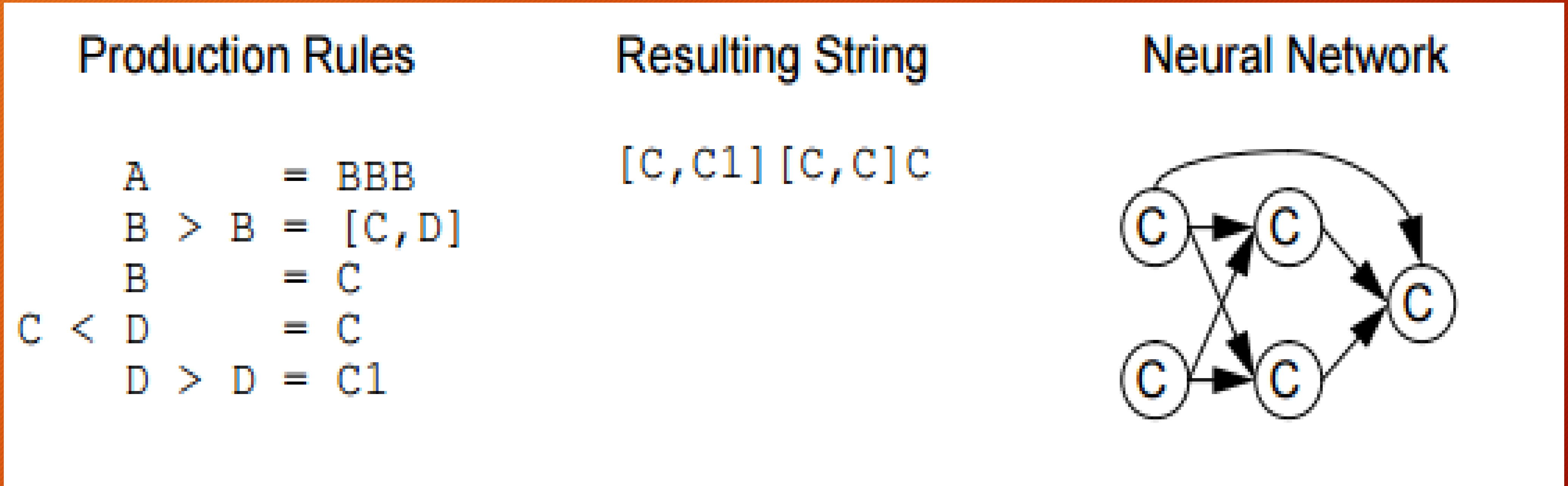
- Three Popular Methods
  - Grammar Encoding Method
    - Makes use of 2x2 Matrix and Symbols to create ANNs
  - Cellular Encoding Method
    - Uses a Tree Whose Nodes Contain Commands for Building the ANN
  - Lindenmayer-Systems
    - Uses Production Rules and Strings of Characters to Build ANN

Indirect Encoding : Cellular Encoding





Indirect Encoding : Lindenmayer-Systems



Production Rules

```
    A       = BBB
    B > B = [C,D]
    B       = C
C < D       = C
    D > D = C1
```

Resulting String

```
[C,C1][C,C]C
```

Neural Network

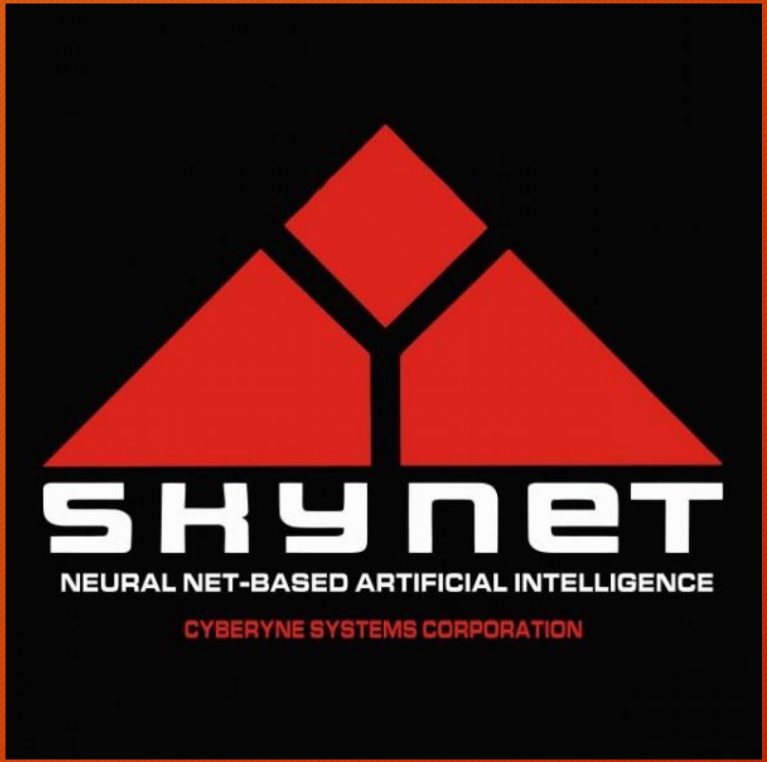
Tricks to Tune your Neuroevolutionary Algorithms

- Spit a Problem Into Individual Subtasks
  - Train on a Subtask Till its Been Mastered
  - Can Allow for Solving Problems That Cannot be Solved Head On
- Biasing the System
  - Introduce Human Knowledge to Guide Evolution
- Anything that Can be Used to Help AANs or Genetic Algorithms
  - Intelligent Mutations can Easily be used on Edge Weights
  - GA methods can Cause “Arms Races” between ANNs
    - This forces ANNs to Elaborate on Existing Behaviors
  - ANN Training Methods can be Used to Further Refine Generated ANNs
    - Even Supervised Methods can Prove Useful
    - Optimized ANNs can be Reintroduced into the Population

Summery

- Neuroevolutionary Algorithms Make Use of GAs and ANNs to Create ANNs through Unsupervised or Reinforced Learning
- Neuroevolutionary Networks have Proven to be Extremely Generalized and Very Resistant to Noise
- Effectiveness of Neuroevolutionary Algorithms Depend Highly on its Encoding
    - Two types: Direct and Indrect
- Neuroevolutionary Networks can Evolve their own ANN Toplogies
- Common ANN or GA Optimization or Learning Methods can be used with Neuroevolutionary Algorithms

NOW GO BUILD


SKYNET
NEURAL NET-BASED ARTIFICIAL INTELLIGENCE
CYBERYNE SYSTEMS CORPORATION