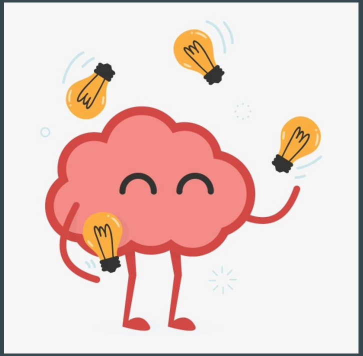
VGP337 - Neural Network & Machine Learning

•••

Instructor: Peter Chan

Neuroevolution

- Neuroevolution is the idea of using evolutionary algorithms to generate artificial neural networks (ANN)
- This can be applied to choosing the number of hidden neurons, the connections, and the weights
- One of the biggest advantages with this method is that neuroevolution can be applied to more problems than supervised learning algorithms, which requires a large, labeled, dataset
- It is also found to be less likely to be stuck in local minima



NEAT

- NEAT is short for NeuroEvolution of Augmenting Topologies and has been developed by Kenneth O. Stanley and Risto Miikkulainen at the University of Texas in 2002
- It applies genetic algorithm to alter both the neuron weights and the structure of a neural network

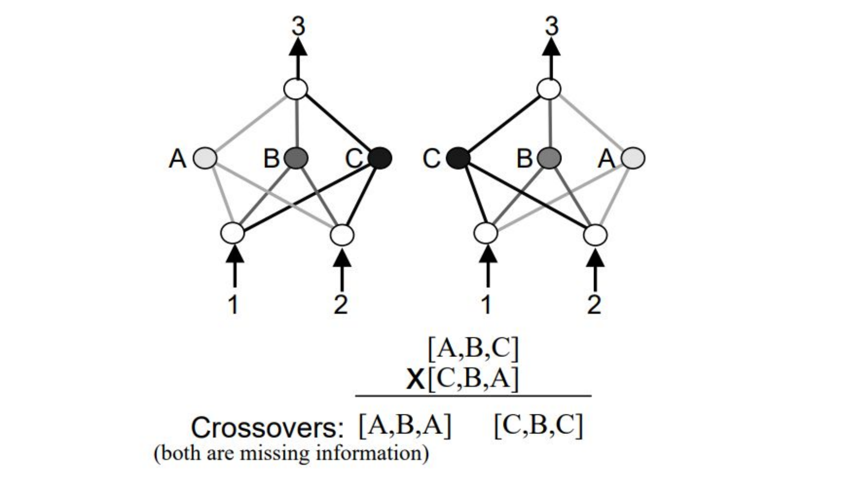
Little Miss Neat

Before NEAT

- Previous neuroevolution attempts mostly only focused on optimizing weights in a chosen fixed network
- The motivation was to remove the dependency on a differentiable model needed for backpropagation (and the need of having labeled data)
- Also, it is non-trivial to apply GA on topologies as it can lead to invalid offspring during crossover

The Competing Convention Problem

- Here are two networks which compute the exact same output



- Since the hidden neurons appear in different order, they will be represented by different chromosomes that are incompatible for crossover

Problems that NEAT Solves

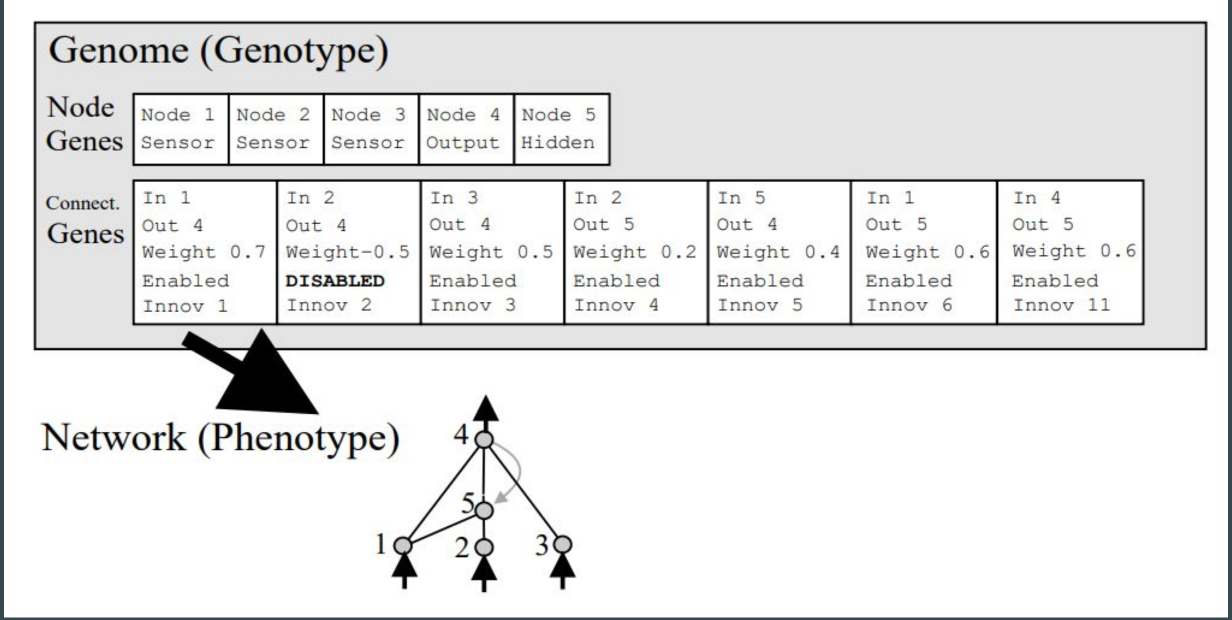
- How can we represent networks with different topologies such that we can apply crossover in a meaningful way?
- How can we protect new topological innovations so that they have a chance to evolve before being eliminated entirely?
- How can we favor minimal topologies so we don’t have explosions on large, complex networks (only pay for necessary complexity)

NEAT Ideas

- Genetic Encoding
- Historical Marking Crossover
- Speciation
- Explicit Fitness Sharing
- Minimal Structure of Initialization

NEAT Encoding

- NEAT encodes a network by using two different genes: a node gene and a connection gene



Genome (Genotype)

Node Genes

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

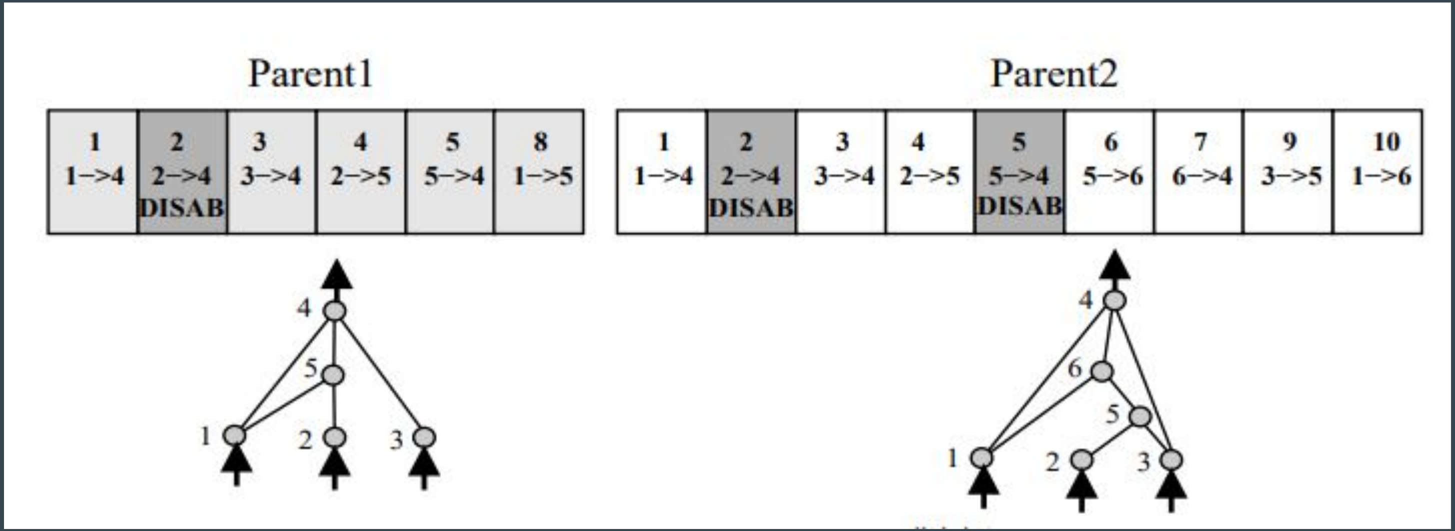
Connect. Genes

In 1	In 2	In 3	In 2	In 5	In 1	In 4
Out 4	Out 4	Out 4	Out 5	Out 4	Out 5	Out 5
Weight 0.7	Weight-0.5	Weight 0.5	Weight 0.2	Weight 0.4	Weight 0.6	Weight 0.6
Enabled	**DISABLED**	Enabled	Enabled	Enabled	Enabled	Enabled
Innov 1	Innov 2	Innov 3	Innov 4	Innov 5	Innov 6	Innov 11

Network (Phenotype)

Innovation Number and Crossover

- The use of innovation number allows us to track genes from different network that represent the same structure



Parent1

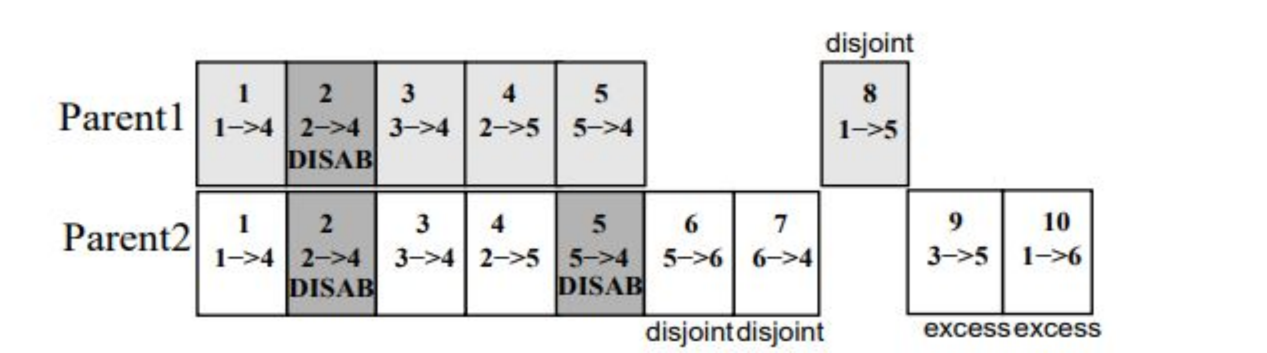
1	2	3	4	5	8
1–>4	2–>4	3–>4	2–>5	5–>4	1–>5
	DISAB				

Parent2

1	2	3	4	5	6	7	9	10
1–>4	2–>4	3–>4	2–>5	5–>4	5–>6	6–>4	3–>5	1–>6
	DISAB			DISAB				

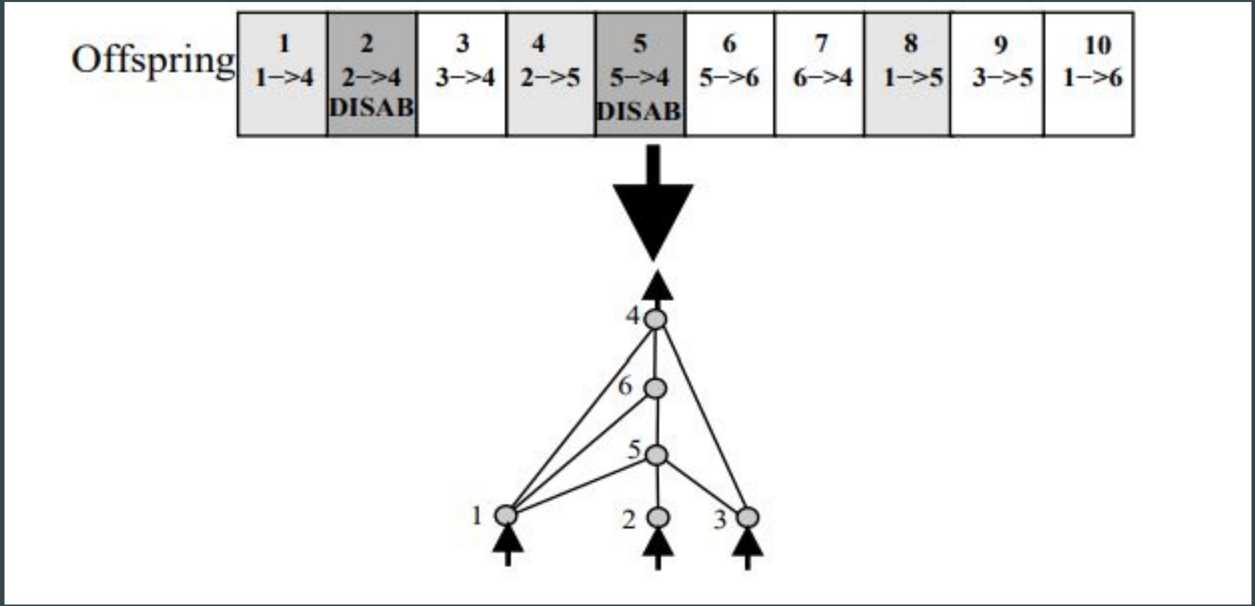
Innovation Number and Crossover

- During crossover, we match up genes from different parent using the innovation numbers



Innovation Number and Crossover

- Matching genes are inherited randomly, whereas disjoint and excess genes are inherited from the more fit parent

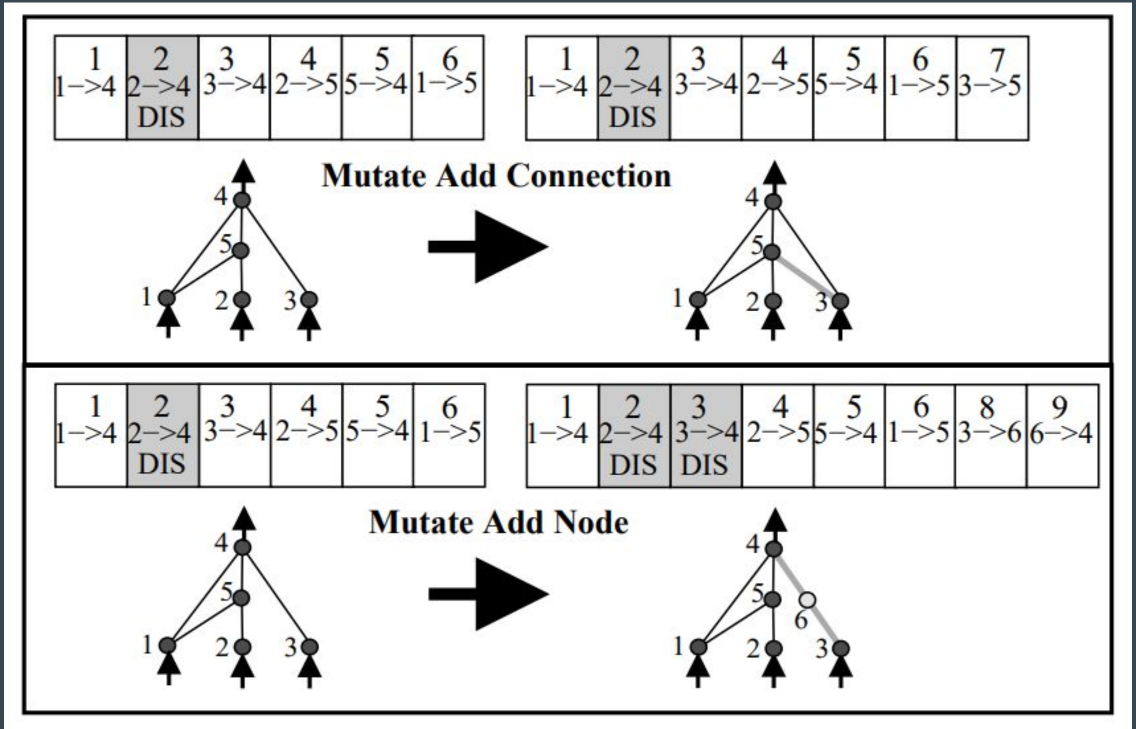


Offspring	1 1->4	2 2->4 DISAB	3 3->4	4 2->5	5 5->4 DISAB	6 5->6	7 6->4	8 1->5	9 3->5	10 1->6

Mutations

- NEAT applies four types of mutations
  - Weight mutation (adjust or randomly pick a new weight)
  - Randomly enable/disable a connection
  - Adding a connection
  - Adding a node

Mutations - Structural



1	2	3	4	5	6
1->4	2->4 DIS	3->4	2->5	5->4	1->5

1	2	3	4	5	6	7
1->4	2->4 DIS	3->4	2->5	5->4	1->5	3->5

1	2	3	4	5	6
1->4	2->4 DIS	3->4	2->5	5->4	1->5

1	2	3	4	5	6	8	9
1->4	2->4 DIS	3->4 DIS	2->5	5->4	1->5	3->6	6->4

Protecting Innovation with Speciation

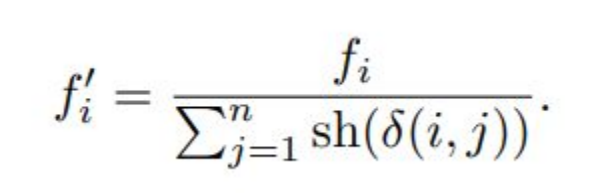
- Networks with new nodes may take longer to optimize
- They will initially have lower fitness and therefore die off during selection
- To protect these innovations, speciation is applied to group compatible genomes so they only compete within their own group
- NEAT applies this formula to measure genome compatibility where:

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \cdot \overline{W}.$$

- *E* is the number of excess genes, *D* is the number of disjoint genes, *W* is the average weight differences of matching genes, and c_1, c_2, c_3, are tuning values

Explicit Fitness Sharing

- To ensure species with organisms that perform well are not overpowering other species, NEAT uses fitness sharing to essentially normalize the fitness across all species
- Here is the formula:



$$f_i' = \frac{f_i}{\sum_{j=1}^{n} \mathrm{sh}(\delta(i,j))}.$$

- The sharing function sh is set to 0 when distance δ(i, j) is above a threshold, otherwise, sh(δ(i, j)) is set to 1

Minimal Structure of Initialization

- NEAT biases the search towards minimal-dimensional spaces by starting out with a uniform population of networks with zero hidden nodes
- New structure is introduced incrementally as structural mutations occur, and only those structures survive that are found to be useful through fitness evaluations
- In other words, the structural elaborations that occur in NEAT are always justified

References

Neuroevolution

Evolving Neural Networks Through Augmenting Topologies

Neuroevolution: A different kind of deep learning

MarI/O

Chapter 11 in Ai Techniques for Game Programming by Mat Buckland