NeuroEvolution of Augmented Topology

William Gebhardt

NAC

Neural Adaptive Computing Laboratory

Week Schedule

Tuesday

- Introduction of Terms
- History of NEAT
- Address upfront challenges of neuroevolution
- Discuss how NEAT addresses these challenges

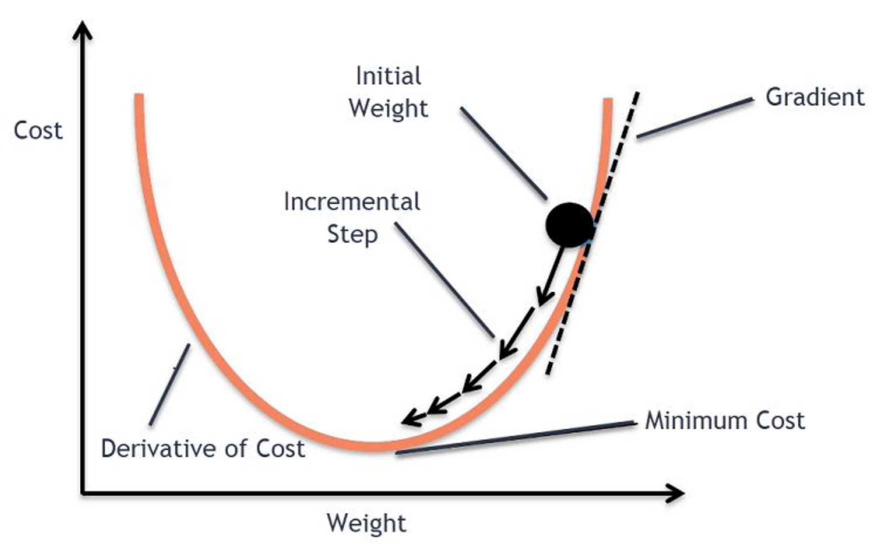
Thursday

- Anything we didn’t get through on Tuesday
- Implementation Walkthrough
- Analysis of a NEAT network on mountain car

Terms and Definitions

Statistical based approaches

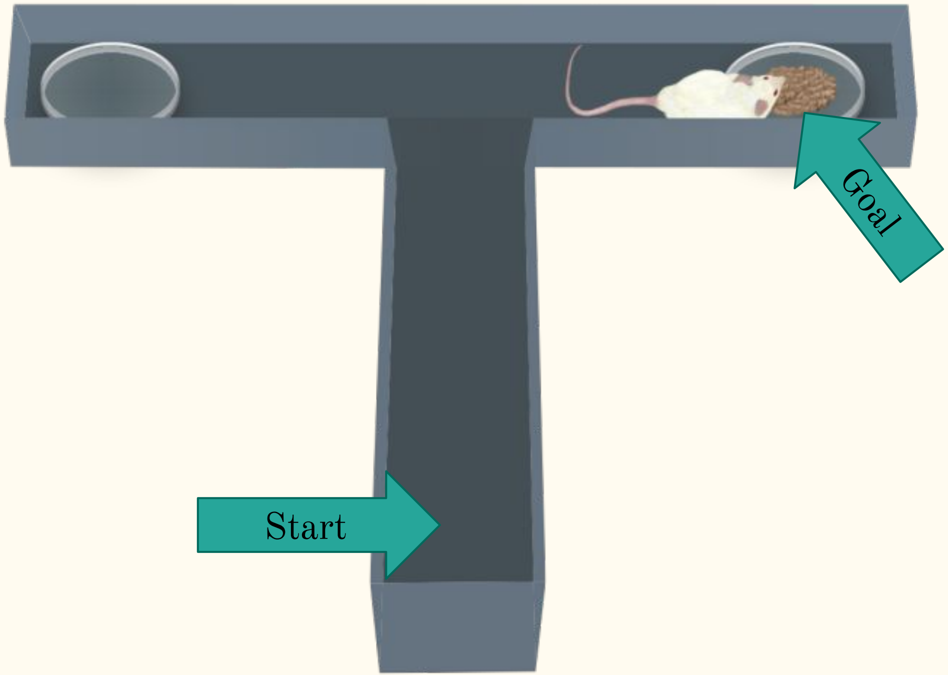
- Gradient Descent
- Purely based on math
- Focuses on marginal improvements
- Single model system



https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/

Reinforcement Learning

- Maximise a reward
  - Rewarded for doing well
  - Punished for failing
- Restarted many time to explore other solutions



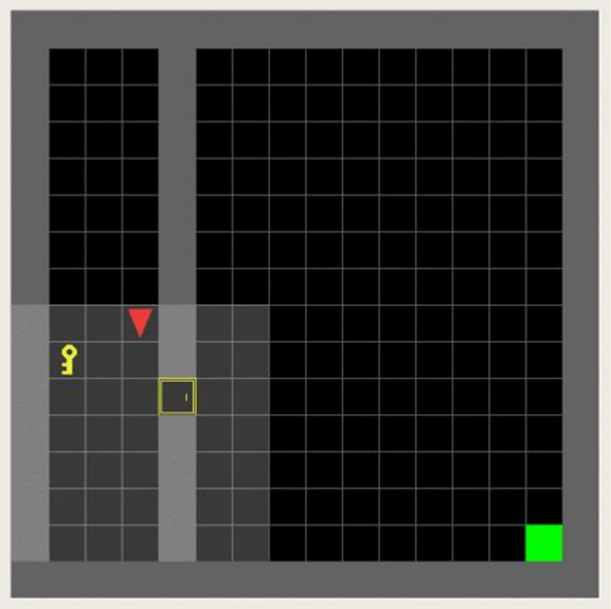
https://en.wikipedia.org/wiki/T-maze

Fitness

- A measure of how performant a model is
- Usually a sum of values for achieving/failing different goals
    - Both positive and negative values

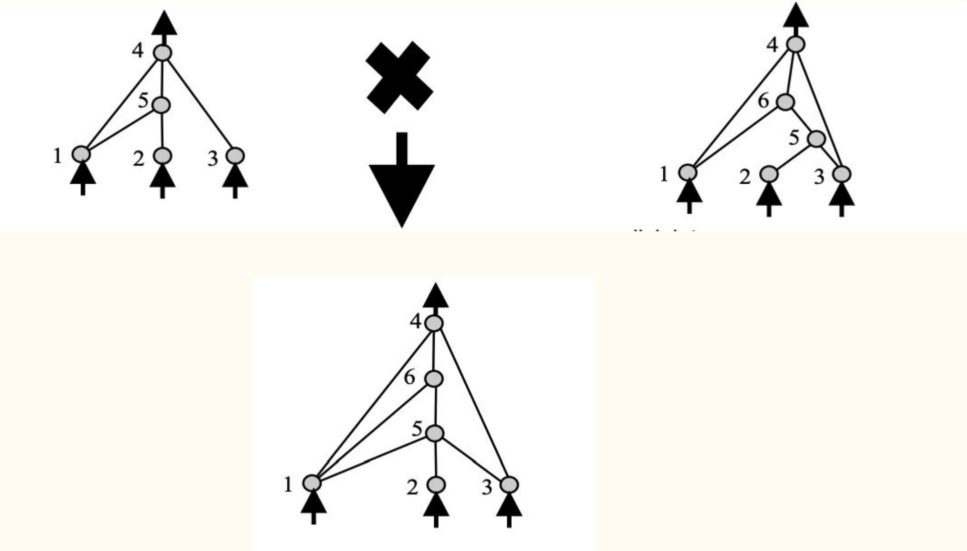
- Positive examples
    - Collecting a key
    - Opening the door
    - Exploring the world
    - Moving to the final goal

- Negative examples
    - Moving over the key without picking it up
    - Never reaching the goal



Crossover

- The combination of two models to produce a new model
- Takes parts of each parent
- One method of adjusting models towards a potentially optimal solution
- Potentially leads to a worse solution

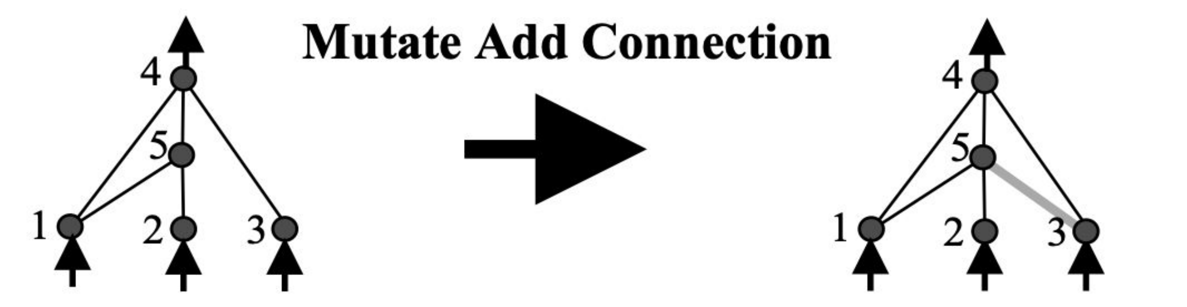


Mutation

- The way new “stuff” appears in models

- The other way models are shifted towards potentially optimal solutions

- Three types of mutations
    - Weight modification
    - Edge Creation
    - Node Creation



NEAT

History

- Coined by Kenneth O. Stanley and Risto Miikkulainen in a 2002 paper
  - Evolving Neural Networks through Augmenting Topologies
  - https://nn.cs.utexas.edu/downloads/papers/stanley.ec02.pdf
- Proposed as an alternative to methods using hand crafted topologies
- Showed massive improvements over static topologies and minor improvements over non-static topologies

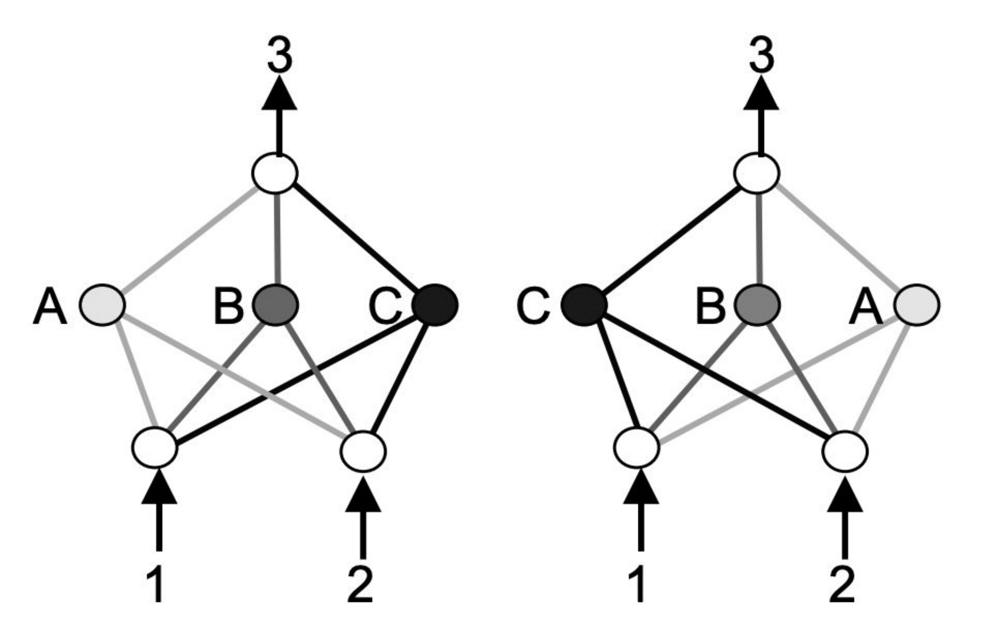
Problems

Encoding The Model

- How to express a model in a way that allows for crossover and mutation to occur
- Some methods
  - Binary Encoding (*structured genetic algorithm*)
  - Graph Encoding (*Parallel Distributed Genetic Programming*)
  - Nonmating (*GeNeralized Acquisition of Recurrent Links*)
  - Indirect Encoding (*Cellular Encoding*)

Competing Conventions

- Many permutations of a set of nodes and edges will express the same network
- For any n hidden units there are $n!$ functionally equivalent permutations.



Innovating Failures

- When new topologies are formed they are likely worse
- Worse structures will have a lower fitness and thus a lower chance of mutation surviving new populations
- Multiple generations are needed to optimize new structure

Solutions & Additions

Genomes

- A template for building an instance of the Genome (Network/Phenotype)

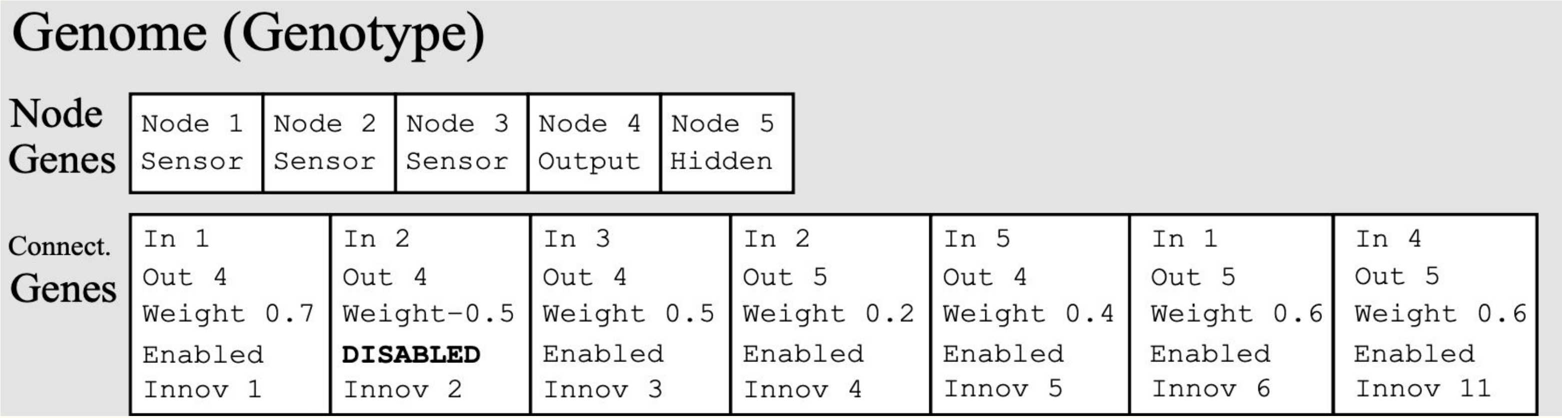
Node Genes

- Input, output, hidden
- Node Id
- Does not guarantee connections

Connection Genes

- Input/output node Ids
- Weight
- Active bit
- Innovation Number

Example Genome



Genome (Genotype)

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

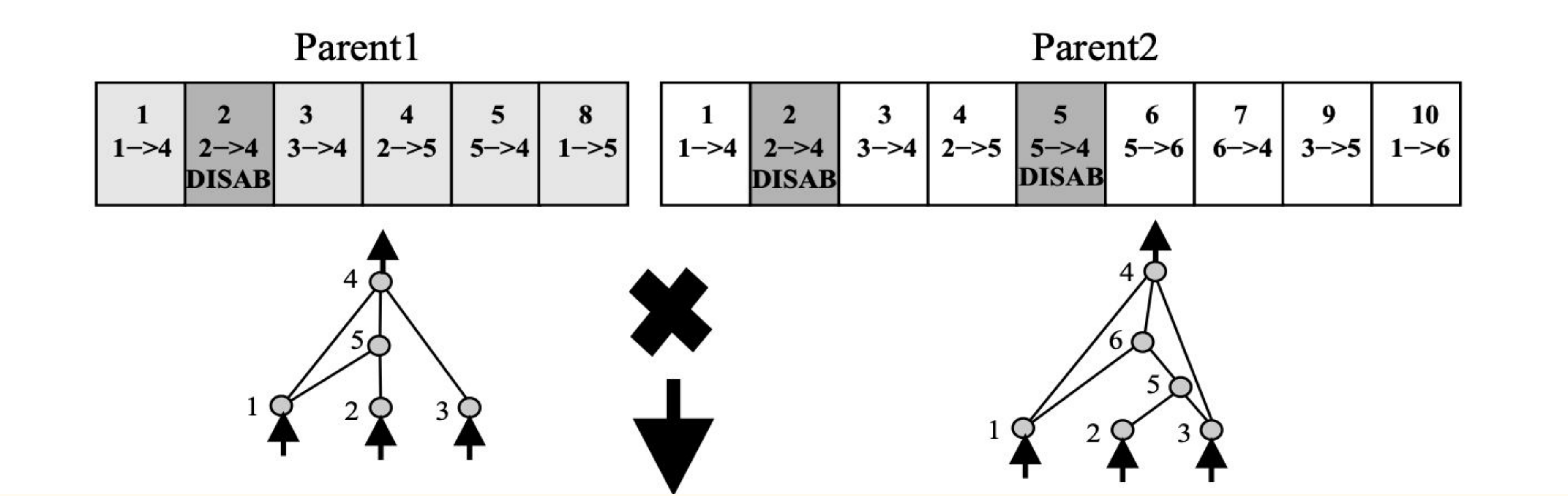
Connect. Genes	In 1 Out 4 Weight 0.7 Enabled Innov 1	In 2 Out 4 Weight-0.5 **DISABLED** Innov 2	In 3 Out 4 Weight 0.5 Enabled Innov 3	In 2 Out 5 Weight 0.2 Enabled Innov 4	In 5 Out 4 Weight 0.4 Enabled Innov 5	In 1 Out 5 Weight 0.6 Enabled Innov 6	In 4 Out 5 Weight 0.6 Enabled Innov 11

Let’s Draw the Phenotype

Historical Markings

- When performing cross over we need to make sure the same genes are crossing over
- Avoid topological level comparisons
- Use innovation numbers associated with each connection gene to align connections
- Global tracker for innovation number
- *Disjoint Gene*: missing the innovation number that is less than the gene's maximum innovation number
- *Excess Gene*: missing the innovation number that is greater than the gene's maximum innovation number

Parents



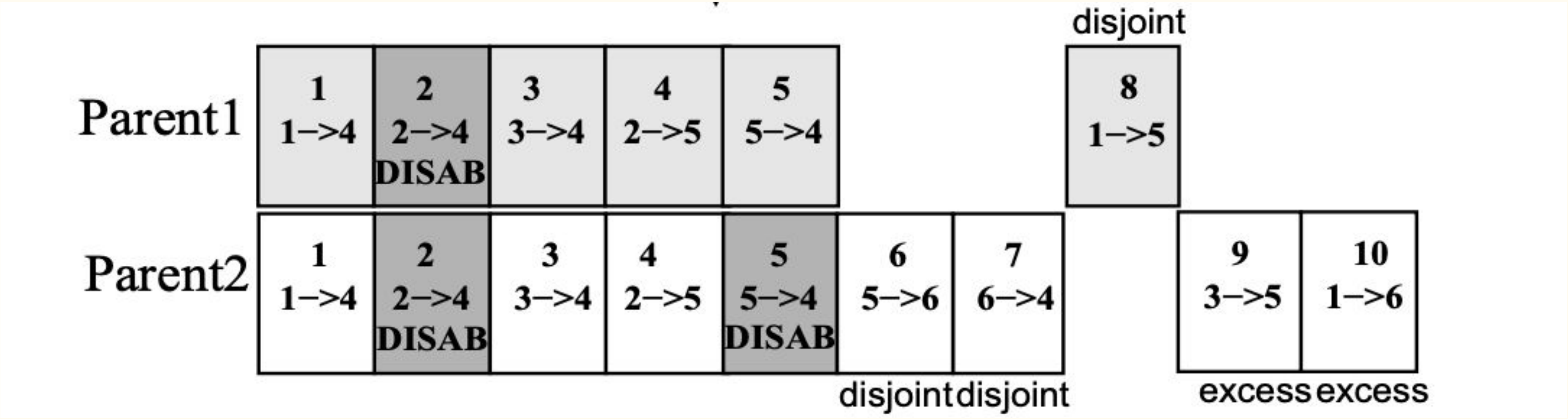
Parent1

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				

Crossover



disjoint

Parent1

1 1–>4

2 2–>4 DISAB

3 3–>4

4 2–>5

5 5–>4

8 1–>5

Parent2

1 1–>4

2 2–>4 DISAB

3 3–>4

4 2–>5

5 5–>4 DISAB

6 5–>6

7 6–>4

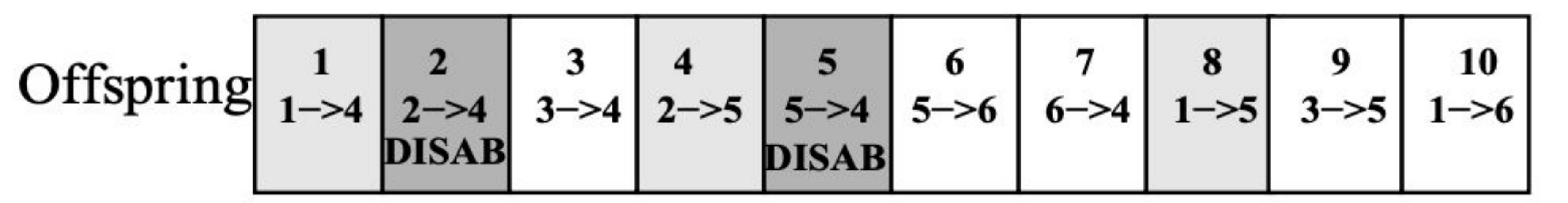
9 3–>5

10 1–>6

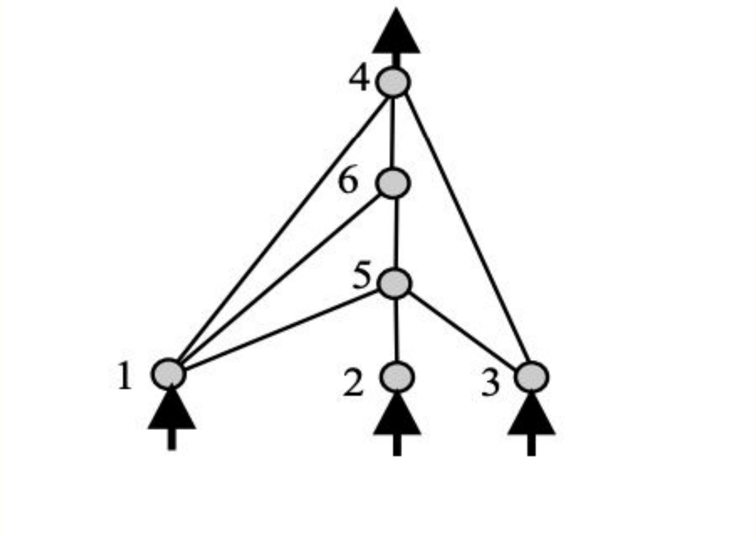
disjoint disjoint

excess excess

Offspring

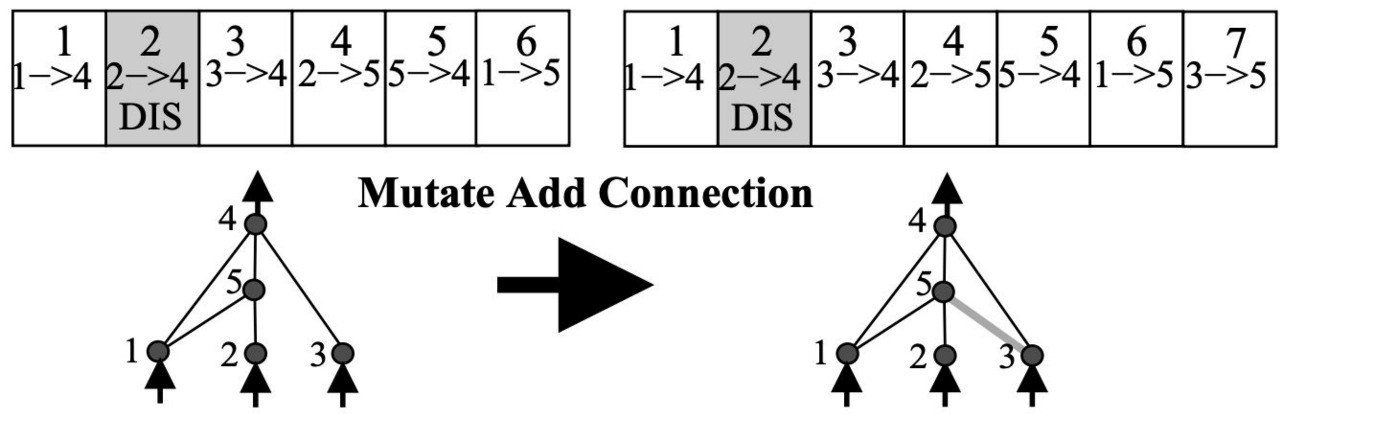


Offspring	1 1->4	2 2->4 DISAB	3 3->4	4 2->5	5 5->4 DISAB	6 5->6	7 6->4	8 1->5	9 3->5	10 1->6



Mutations - Connection Gene

- Find an edge that currently does not exist in the genome
- Add the connection gene to the genome with a random weight and next innovation number



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

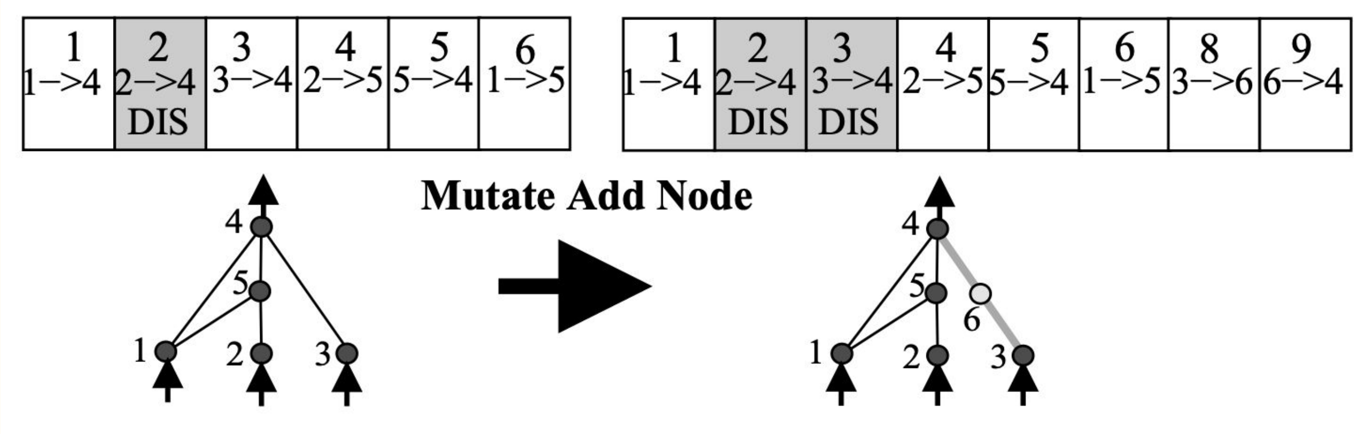
1	2	3	4	5	6	7
1–>4	2–>4 DIS	3–>4	2–>5	5–>4	1–>5	3–>5

Mutations - Node Gene

- Take a currently existing connection and disable it

- Add a node gene to the genome

- Add to connection genes to the genome to rebuild the previous edge



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

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

Speciation

- Used to protect suboptimal species from extinction
- Uses a defined compatibility distance and threshold to divide up genomes
- Competition for crossover primarily happens inside of species

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \bar{W}$$

E: excess connections

D: disjoint connections

W: average weight distances between matching genes

N: number of genes

c1, c2, c3: weighting factors

Species Selection

- Each species has a representative from the previous generation
  - Chosen randomly from amongst the genomes
- Each new genome is compared in order via the compatibility distance
- Genomes are assigned to the first species their compatibility distance falls under a certain threshold
- If none are met it will be a new species and be added to the set of species in the population

Fitness Sharing

- Competition primarily happens inside a species through *explicit fitness sharing*
- sh() is 1 if below the threshold and 0 everywhere else
- Effectively this normalizes fitness inside each species
- After this normalization genome selection for the next generation proceeds as normal

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

Thoughts for Thursday

- Think about how you would construct a Genome and generate a Phenotype from it
- Think about how you would do mutation and crossover efficiently
- How would you simulate many different topologies at the same time during training?