**NeuroEvolution of Augmenting Topologies (NEAT)**

NeuroEvolution of Augmenting Topologies (NEAT) is a genetic algorithm (GA) for the generation of evolving artificial neural networks (a neuroevolution technique) developed by Ken Stanley in 2002 while at The University of Texas at Austin. It alters both the weighting parameters and structures of networks, attempting to find a balance between the fitness of evolved solutions and their diversity. It is based on applying three key techniques: tracking genes with history markers to allow crossover among topologies, applying speciation (the evolution of species) to preserve innovations, and developing topologies incrementally from simple initial structures ("complexifying"). (https://en.wikipedia.org/wiki/Neuroevolution\_of\_augmenting\_topologies)

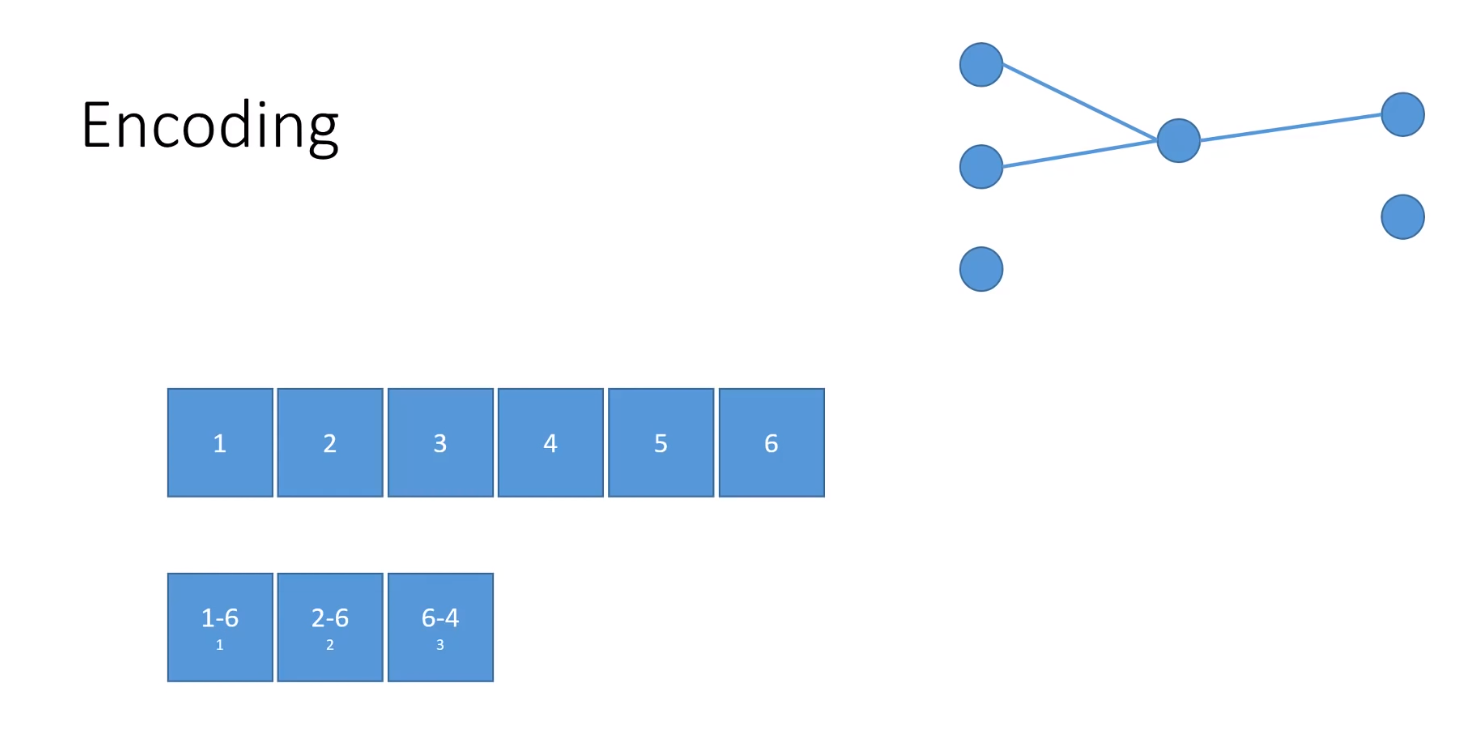
Evolution: Selection, Crossover, Mutation

Neuro: neural network

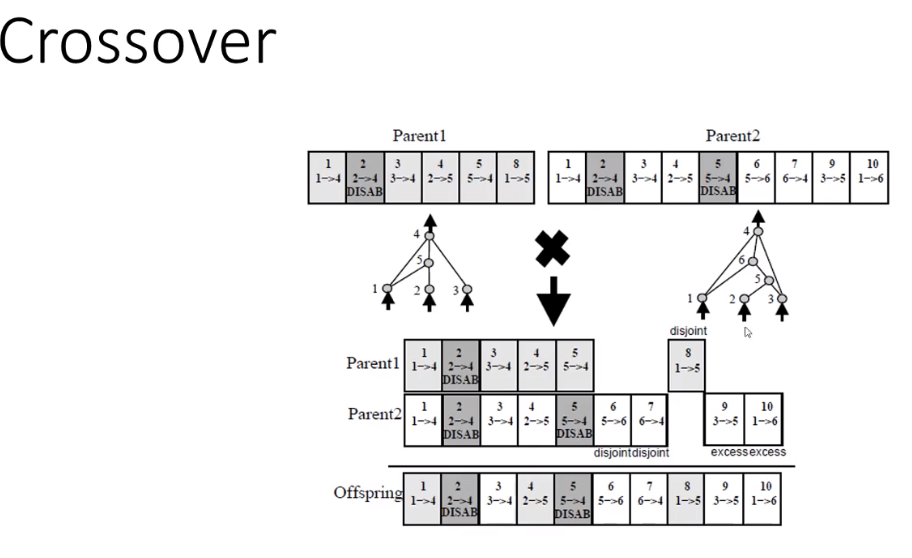
Augmenting Topologies: the amount of connections and nodes in the network will be vary (we have to start small, therefore it will take too long)

Since the number of hidden layers and nodes vary, there is no weight array that is used to do genetic mutation with the previous version, therefore no crossover from the weight arrays.

**Encoding**: uses genes, a genome is a set of genes. Two different type of genes. Node genes and connection genes. Node genes contain the identification of the node. Connection genes contain the node from, the node to and the innovation (identification) number. Innovation numbers are very important, it determines when a connection has been created, it is incremented over time. The connection genes stay the same across networks if the connection is the same.



**Cross over**: The connection genomes between two parents are compared. (each gene also contains the weight and a Boolean value that represent if a gene is disabled or not) If both genomes contain matching connections, one will be chosen randomly for the offspring, if one genome contains a connection that the other genome does not have (disjoint), it will go to the offspring. Finally, we have excess genes. If one genome has more genes at the end of the genes list, they will carry over only if that specific parent is fitter than the other.



**Mutation**: 5 different types

Mutate\_link: add a new link with random weight between -2 and 2. If the connection already exist, no new innovation number, if new 🡪 new innovation number

Mutate\_node: add new connection in a link, setting the weight of the from link to 1 and the to link to the value of the original link. The new node is getting a new innovation number, the connections too.

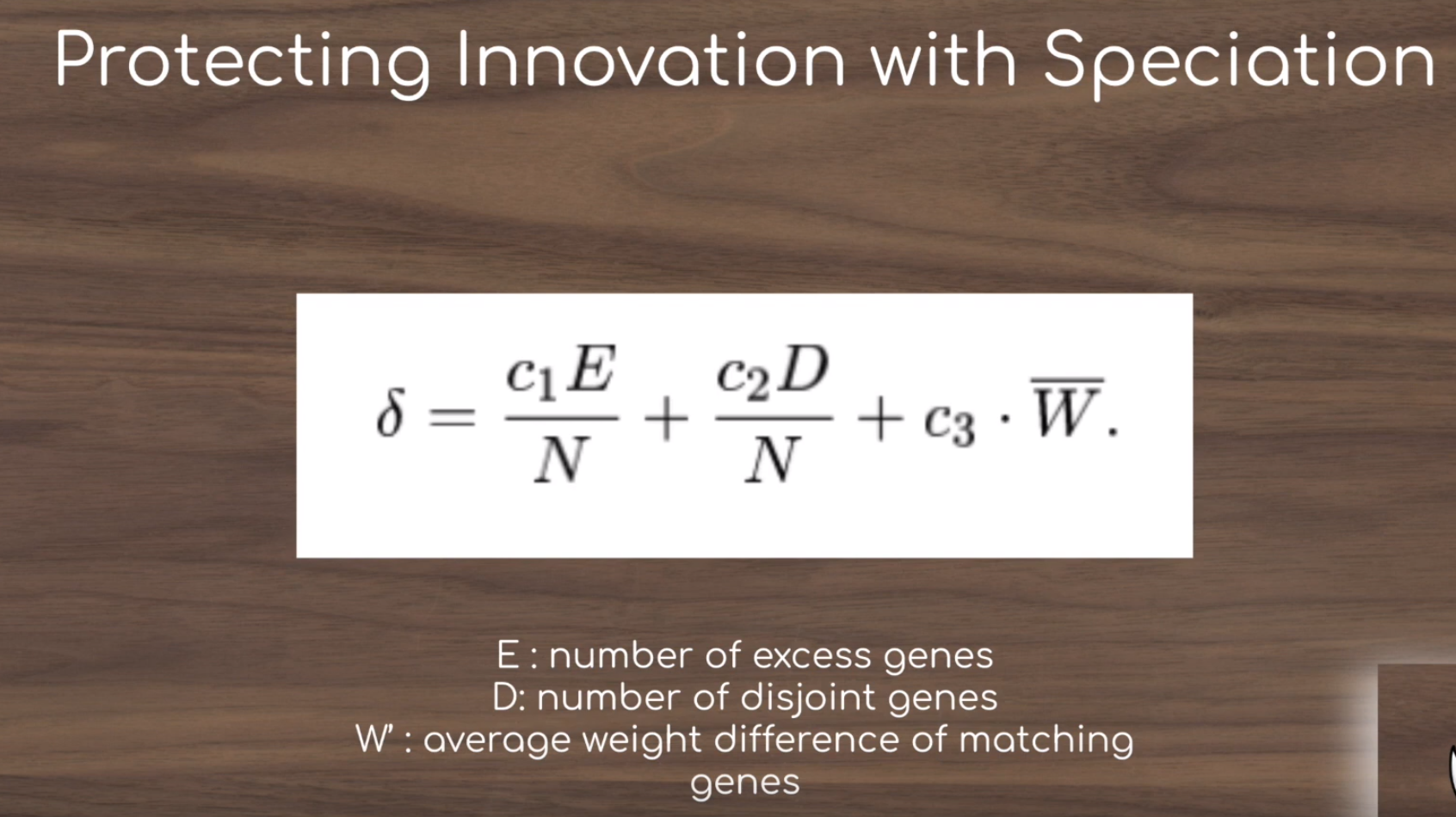
Mutate\_enable\_disable: we can enable/disable a random connection a percentage of the connections

Mutate\_weight\_shift: take a random weight and multiply by a number between 0 and 2

Mutate\_weight\_random: reselect a random weight for a random connection

**Selection**:

NEAT uses a function that groups neural network that have a similar structure so that they don’t compete against other neural networks with different structures. This process allows neural networks with a lower starting fitness the ability to develop and maybe become the fittest neural network. Neural networks in a certain threshold are grouped together in a specie.



Sources:

* NEAT presentation 1:<https://www.youtube.com/watch?v=VMQOa4-rVxE>
* NEAT presentation 1:<https://www.youtube.com/watch?v=b3D8jPmcw-g>
* Mario : <https://www.youtube.com/watch?v=qv6UVOQ0F44>
* <https://en.wikipedia.org/wiki/Neuroevolution_of_augmenting_topologies>