Neuro Evolution of Augmenting Topologies NEAT

Iheb Gafsi*

INSAT Student

iheb.engineer@gmail.com

Introduction:

Reinforcement learning is one of those fields in which we don't supervise our model's performance so we can't just optimize our parameters based on prediction versus label loss. But instead, we train the model based on what we want them to do, and what I mean by that is specifically giving it rewards when they perform what we want them to do, like when a self-driving car makes a good turn or when a player in chess checks a king… and we punish the model when it doesn't do the right thing.

Neuroevolution (NE), has shown a promising performance in this field of reinforcement learning. Working with genetic algorithms, breeding the best-performing model until we reach the best neural network, makes it much easier and faster to reach the point we wanted to reach than the traditional reinforcement learning algorithms such as Q-Learning as NE learns behavior instead of a value function. However, in traditional NE algorithms, we have a fixed topology and we just want to adjust the weights and biases to enhance the model's performance. This does not lead to an optimal, flexible, scalable, and adaptable solution.

Thus, the Neuroevolution Augmented Topologies (NEAT) algorithm came up with a solution that uses different topologies genotypes and sees how they perform every generation, applying natural selection, crossover, mutation, and speciation to eventually reach an optimal solution.

Mechanism:

Genetic Encoding:

Unlike traditional genetic algorithms, in NEAT we will have two types of genes, node genes and connection genes. Node genes are going to be described by their IDs, biases, sensor or output, or hidden property, and the corresponding activation function. Connection genes on the other hand are described by their weights, input nodes, output nodes, and the enabled-disabled property, but most importantly identified by their innovation number which is going to help us in other tasks.

A genome or genotype is a collection of both of these genes and its corresponding neural network is its phenotype. This is an example of a genetic encoding:

ID= 0 SENSOR	ID= 1 SENSOR	ID= 2 OUTPUT	ID= 3 OUTPUT	ID= 4 OUTPUT	ID= 5 HIDDEN	ID= 6 HIDDEN

INN: 8 Input: 0 Output: 2 ENABLED	INN: 24 Input: 0 Output: 4 ENABLED	INN: 34 Input: 1 Output: 4 ENABLED	INN: 23 Input: 1 Output: 3 **DISABLED**	INN: 47 Input: 1 Output: 5 ENABLED	INN: 62 Input: 1 Output: 6 ENABLED	INN: 61 Input: 5 Output: 2 ENABLED	INN: 75 Input: 5 Output: 3 ENABLED	INN: 138 Input: 5 Output: 6 ENABLED	INN: 93 Input: 6 Output: 3 ENABLED	INN: 114 Input: 6 Output: 4 ENABLED

Note that in this genetic encoding, I used the Cantor pairing function to get the unique innovation number for every gene, it takes the ids of the input node and output node and spits out a unique identifier:

$$\pi: \mathbb{N}^2 \to \mathbb{N}$$
$$\pi(n, m) \to (n + m)(n + m + 1) + m$$

Figure 1 down below illustrates in a graph the corresponding phenotype to our example:

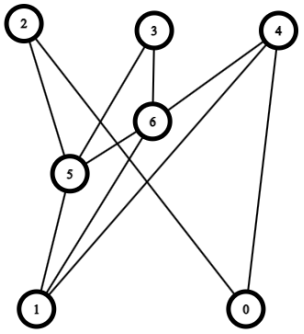


Figure 1: Corresponding phenotype to our table

Natural Selection:

These genomes that we create will form a population, they're going to compete to survive that's why we will associate every genome to fitness. The fitness is a score that determines how well the genome performs, we should just make a fitness function that evaluates the genome. For instance, we can make a fitness function that adds 5 points to the fitness of a snake that eats an apple and subtracts 2 if it hits an obstacle.

Genomes with the highest fitness scores will survive and will breed to repopulate while others will die. Natural selection is a building block in the NEAT algorithm and it is inspired by real natural selection.

"Survival is for the fittest"

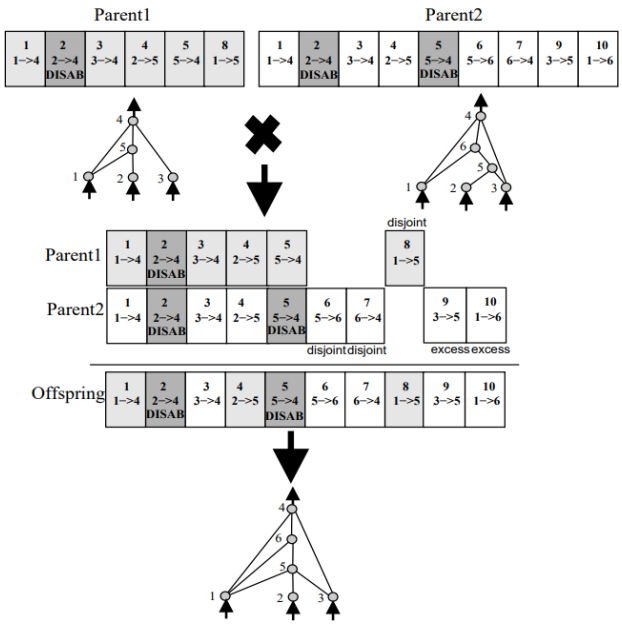
On the Origin of Species – **Charles Darwin**

Crossover:

When repopulating the population of genomes, we will crossover two random genomes every time to get an offspring genome that will take the light later to compete with the others. Usually in genetic algorithms, we just fuse the weights and biases of the winning genomes till we find some phenotype that has the best weights and biases. However, this is not the case in NEAT as we try to find the most optimal solution with the best weights and biases, therefore it's going to be trickier to breed two genomes with different structures. But here's how it works:

1- Inherit all mutual nodes from one parent or the other randomly
2- Inherit the excess and disjoint nodes from both of the parents
3- Align connection genes with the same innovation number
4- Inherit all mutual connections from a random parent every time
5- Inherit the excess and disjoint connections from the fittest parent

Here's an example from the original paper by Kenneth Stanley:



Mutation:

After getting an offspring genome or simply generating a new one we should mutate them to embrace the diversity and get different performances. When breeding the fittest genomes without mutations, our model won't change but instead keeps itself at a constant range of fitness scores. So, we give slight changes to the offspring that could cause it to have either a new talent to beat the others or a disability that will cause it to suffer and die. In regular genetic algorithms, we just either change one of the weights or add small perturbations but there's a genius idea behind NEAT's mutation it's that we can add, connections and nodes, perturb or change weights and biases, or enable/disable connections. This will maximize the diversity of our phenotypes. Let's give an example of every mutation type:

- **Perturbing weight or bias:**
  In this type of mutation, we just add a random small number, generally between -1 and 1 to a random connection weight or a random node bias.
- **Changing weight or bias:**
  In this type of mutation, we change the value of a weight or a bias of a connection or a node.
- **Adding a connection:**
  Adding a connection mutation can only happen when the corresponding neural network is not fully connected. In that case, we choose two random nodes and connect them in a way we don't lose the one-directional pattern of our neural network, the weight is going to be arbitrary. **Figure 2** illustrates an example of this mutation where we add a connection between node 5 and node 2:

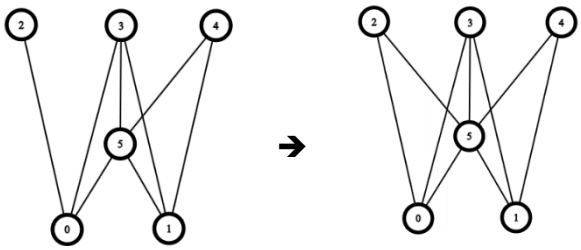


Figure 2: Adding Connection Mutation

- **Adding a node:**
  Similar to adding a connection mutation, we choose an arbitrary connection and put a new node in its center, the new node is going to have the bias of zero and the old connection is going to be divided into two connections, one connects the input node to the new node with weight 1 and one will connect the new node to the output node. This won't make any changes initially but when weight/bias perturb/change mutations or crossovers happen, this will be effective. **Figure 3** shows an example of adding a new node in the connection connecting node 5 to node 2.

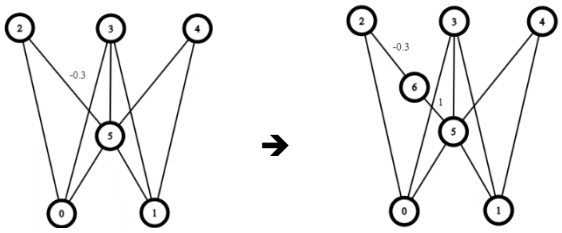


Figure 3: Adding Node Mutation

"Darwinism is not a theory of random chance. It is a theory of random mutation plus non-random cumulative natural selection"

The Blind Watchmaker – **Richard Dawkins**

Speciation:

Natural selection does get us the best solutions but using it alone approaches the solution from one direction so the best idea is to classify our population of genomes into species containing the most similar genomes. This similarity is going to be determined by the genomic distance function δ which I will be using as:

$$\delta = c_1\overline{ED} + c_2\bar{B} + c_3\bar{W}$$

Where $\overline{ED}$ is the excess/disjoint average, $\bar{B}$ is the bias difference average, and $\bar{W}$ is the weight difference average.

A genome belongs to a species if the genomic distance between it and a random genome from that particular species is less than the delta threshold that you will fix. The higher the fitness threshold you put the fewer species you get.

Conclusion:

Neuro Evolution of Augmenting Topologies has successfully shown the best performances at reinforcement learning tasks which unlocks the door for more innovation and development in the future.

It's always efficient to take inspiration from nature, make planes inspired by birds, and make NEAT inspired by evolution theory.