NEAT Notes

**Protecting Innovation**

* Usually, adding new structure initially causes the fitness of a network to decrease.
* This is usually fixed with Speciation
* Can it be fixed instead by setting default values such that the new structure causes no change in the fitness, but unlocks better potential, by enabling further training of the new non-linearity. E.g. adding a new node in a connection, which propagates the same information as before.
  + This may require changing how weights are mutated. Not by random, but by incrementing/decrementing them.
  + This organism may still be very insignificant in the population and thus die out by chance. But this factor could be used to help prevent against over engineering, and promote minimalist topologies.
  + “The new connection leading into the new node receives a weight of 1, and the new connection leading out receives the same weight as the old connection. This method of adding nodes was chosen in order to minimize the initial effect of the mutation.” – MIT paper

**Preventing Same topological innovations from having different innovation numbers**

* Create a list of all the topological innovations that occur in the same generation to ensure each identical mutation is assigned the same innovation number. – *MIt PAPER*
* I propose taking this idea further and making this list of topological innovation persist throughout all generations.
* This would result in each genome having an unsorted list of innovation numbers. It may be necessary to sort this list, to simplify/enable the crossover process.
* We cannot take this further by making genomes ONLY a list of these innovation numbers, as each genome trains it’s own weights, thus needs to store this weight information with its corresponding innovation number. Different combinations of innovation numbers may require different weights for the same connections, thus I currently believe each genome needs to store its own weight information. (if I can find a way around this it may be much faster)

**Encoding structural information in the Innovation number**

* The encoding will depend on the number of inputs and outputs (and biases) being constant, throughout training.
* Will not have to have a globally incrementing counter for each innovation
* E.g. connection from node 1 to 3 could be encoded by finding the hash of each number and applying some crossover, or appending. Using VERY basic hashing algo. Doesn’t have to be safe.
* INNOVATION NUMBER AS TUPLE? (input node, output node)