Nawaf Abdullah

PHYS 497 – Project 1

**Abstract:**

For this assignment, a set of “neural networks” are randomly constructed by a program written in Python. Each network node is assigned a set of random actions to be performed on its input, which are selected from a larger set of all the possible actions. Those networks are then subjected to an artificial selection process that returns the best network at accomplishing a user-set criterion. The idea behind this process of networks artificial selection is that it can be used to optimize for the best solution, where outcome of interactions between system components is unknown. *The purpose of this paper is to describe the network that is used in the software while it is currently under development.*

**Network Definition Overview:**

Nodes: Each node in the network contains a mathematical function that is randomly generated. After taking a numerical input, the mathematical operations performed on it is determined completely random.

Edges: The edges serve as a link between the nodes, determining which Node B the data in Node A will be transferred to for processing again in Node B. For this initial phase of the project and due to the complexity of implementation and time constrains, the edges are not weighted nor directed. However, in theory, a weight can be assigned to edges between neuron affecting the node interactions and the mathematical operations performed on each node data.

The System: The system takes an input at one it’s nodes and start by processing it there. The output returned by is then sent to an adjacent node for processing. This continues from node to node within the graph until all nodes have been visited. After that, the graph returns a net value . A graph can generate a different result, given the same previous input, when running the whole process again. This is due to the random selection of the adjacent node which the value is passed to from (See Figure 1).

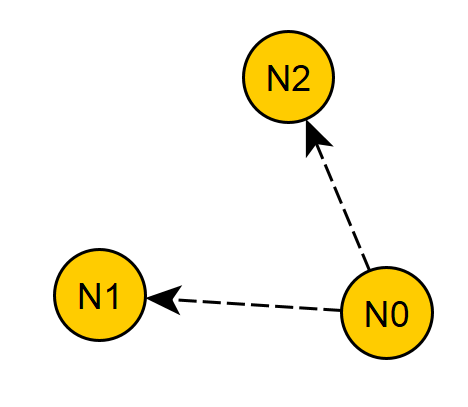


Figure 1: Two possible adjacent nodes to pass ’s value to. One of them is selected randomly. The arrows on the edges are only for illustration purposes, edges are not necessarily directed.

After multiple graphs have been generated, they are passed in a list to an “Environment” class, which then returns optimized networks for the user-set selection criterion.

**Tools used:**

Python: version 3.6 was used to code the project.

NetworkX: The Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. (Figure 2)

Project is hosted on Github: <https://github.com/MentalN/Genetic-Network>

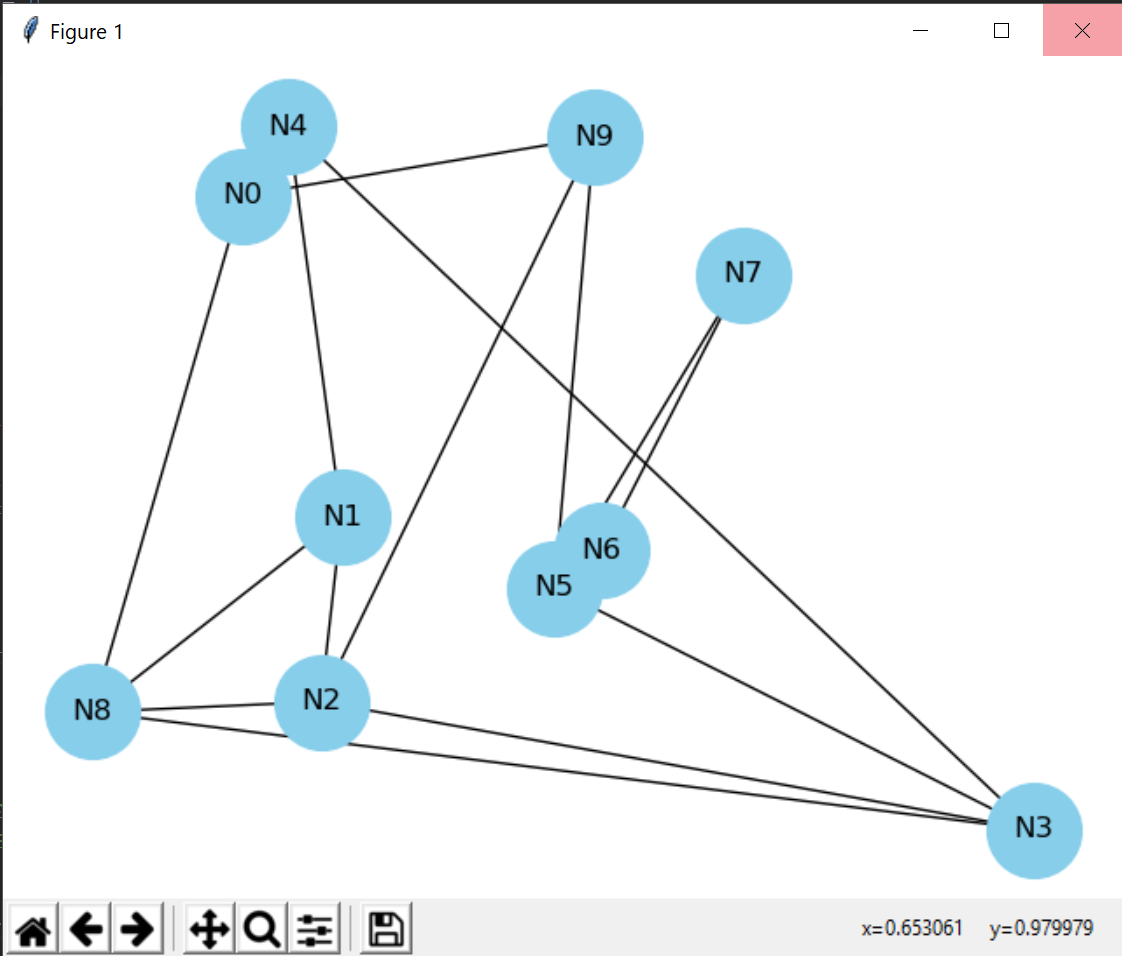


Figure 2: An example of a graph plotted by the program using NetworkX Python Package. This graph is composed of 10 Nodes. Each node has 4 undirected edges.