A graphical application to simplify the use of NEAT to evolve neural networks to control video game characters.

Venetsia Tsvetomirova Krasteva

Submitted in partial fulfilment of
the requirements of Edinburgh Napier University
for the Degree of
BENG(HONS) SOFTWARE ENGINEERING

School of Computing

December 2021

Authorship Declaration

I, Venetsia Tsvetomirova Krasteva confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the Source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines

Signed: *Venetsia*

Date: 01/12/2021

Matriculation no: 40313507

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below *one* of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

Abstract

The purpose of developing this program is to make NEAT (NeuroEvolution of Augmenting Topologies) more accessible to users. The NEAT algorithm is used to evolve artificial neural networks to solve control problems. The program's main aim is to be educational so people with no artificial intelligence experience can go through the education mode and understand more about using NEAT. The aim is to enable the user to edit the configuration file with ease, train a neural network to play a game and see the results. Currently, getting NEAT to work with any easy-to-use suite of reinforcement learning tasks library has many compatibility and integration issues.

The final prototype is a working program that any user can load on their machine, edit the configuration file, train a neural network, see the progress within checkpoints and see the found winner of the neural network. The games included in the program are indeed working as each of the games were tested and then added. Changes to the NEAT algorithm were made to include specific features so users can compare the progress on a particular generation with another or final neural network. Education mode has all of the necessary lessons starting from the basics of Artificial Intelligence and going through the necessary topics before finally learning more about NEAT.

The program is evaluated by users who have volunteered to test the program on their machine or use remote control and give feedback answering a few questions that aim to give the developer an idea of what needs more improving or changed.

Contents

List of Tables

List of Figures

Acknowledgements

I would first like to acknowledge Dr Simon Powers for the incredible support, guidance and advice in the beginning and in the weekly meetings throughout the development of this project as this program and report would not have been possible to achieve without the provided help. I would also like to thank the staff of Napier University's School of Computing for providing me the experience, knowledge, and encouragement to attempt the project.

I would also like to thank the second marker of the project, Dr Andreas Steyven , for his suggestions at the interim meeting.

I would also like to express my thanks to Kenneth O. Stanley for developing the NEAT algorithm and to the Open AI Gym library developers for providing such a great toolkit that included many game environments.

I would like to thank my loving boyfriend who kept encouraging and enduring me throughout all the late nights that I spent developing and writing the project.

I want to thank my family for giving me a chance to study abroad and supporting me thoughout my years of education.

Finally, a special thanks to all of those who completed questionnaires from the survey for the evaluation.

1 Introduction

1.1 Project Background

Many people have been fascinated with artificial intelligence (AI) and machine learning algorithms (ML), and in recent years, AI has become more engaged in our day-to-day lives. Most of the technology we use uses some AI tool to make our lives easier and improve them. For example, Siri uses voice queries and a natural language user interface to function on iOS. Cortana is another AI app that comes in handy as a virtual assistant on several devices and environments. Socratic is an AI app that uses text and speech recognition. It aims to help students with math and other homework. AI has been integrated into emotional health apps to help users take control of their emotional health. These applications are used to help us get better and work better.

Another use of AI is to make it accessible for analysts, researchers, and others with no coding experience for some of the learning methods, making it accessible to anyone to use without needing writing a single line of code, making it much easier for the person to perform research or job. For example, Microsoft has developed Lobe, making machine learning accessible for everyone to develop a classification model. Unfortunately, these applications are still relatively new, so few tools enable accessibility and ease of use.

Having an application that people can use to train a neural network for control without using code would be another accessible tool that people could benefit from as, to this day, there is not an easy-to-use application that does need no programming knowledge. That is why this project focuses on filling the gap on building such an application to control a video game character with an evolved neural network in a simplified graphical application that would be accessible without coding experience. In addition, the program would allow the users to learn more about AI, NEAT and how to use it within the program with Education mode, which will extend the standard program.

1.2 Motivation

Artificial Intelligence is an important field that tries to create a computer that thinks and acts like a human and tries to resolve issues and solve problems differently by searching for patterns, abnormalities, or trains on experience

(rewards). Artificial Intelligence and all its subfields have been integrated into many tasks that, for example, a human would be dangerous to perform or improve performance.
Different platforms enable you to use machine learning without a single line of code that lets you do a wide range of tasks from image classification to natural language processing like CreateML, Google AutoML, MakeML, Teachable Machine and Lobe Microsoft.

1.3 Project Aims and Objectives

This project aims to build an application accessible to users who do not have any coding experience and educate them. In order to make the algorithm accessible, the program should allow the user to train a neural network using NEAT to control a game character in multiple Atari games. The project is to be developed as an easy-to-use graphical application.

The project objectives:

1. Produce a literature review of research on the evolution of neural network characters for video games focusing on the NEAT library.
2. Design the front-end (graphical user interface) for the application.
3. Design the software architecture, i.e., how the application connects with NEAT and the video game.
4. Implement the application.
5. Test the application on a variety of Atari video games.
6. Produce the dissertation document detailing the above work.

1.4 Report Structure

Chapter 1 Introduction:

Describe the project's aim and answers to why we chose this project and a quick overview.

Chapter 2 Literature Review:

Background and explanation of Neural Networks and the library this project use on a specific domain.

Chapter 3 Requirements Engineering:

Explains the process of establishing and managing requirements in order for the application to be successful

Chapter 4 Design and Implementation:

Clarifies the approach taken to implement the program, the application's initial design, the applications' functions, and the testing conducted.

Chapter 5 User Evaluation:

Feedback from conducted research on prototype

Chapter 6 Conclusion:

An overview of the tasks achieved in this project, analysis of the program developed and any discussion of possible future developments

2 Literature Review

2.1 Artificial Intelligence

Artificial Intelligence (AI), according to Russell & Norvig (1995, Chapter 1), attempts to understand intelligent agents. The dimensions in AI are concerned with the thought process, reasoning, and behaviour. These two dimensions are split into four approaches, where thinking and acting humanly is not equal to thinking or acting rationally. Human-like behaviour measures human performance, whereas rationality is a concept that does the right thing to look for the ideal performance, given what a machine knows at a certain point.

2.1.1 Artificial Neural Networks

According to a review (Kumar & Thakur, 2012), Artificial Neural Networks (ANN) are AI branches. ANN's are inspired by the human brain, where the neuron is the part of a nerve cell that is the fundamental functioning unit of all nervous system tissue. Each neuron forms synapses from a dozed to a hundred thousand other neurons, and signals are propagated from neuron to neuron. [See **Fig 1**] The neural network is based on the same analogy. It is essentially several nodes that form links, where each link has a numeric weight associated with it that the learning element can modify. [See **Fig. 2**]

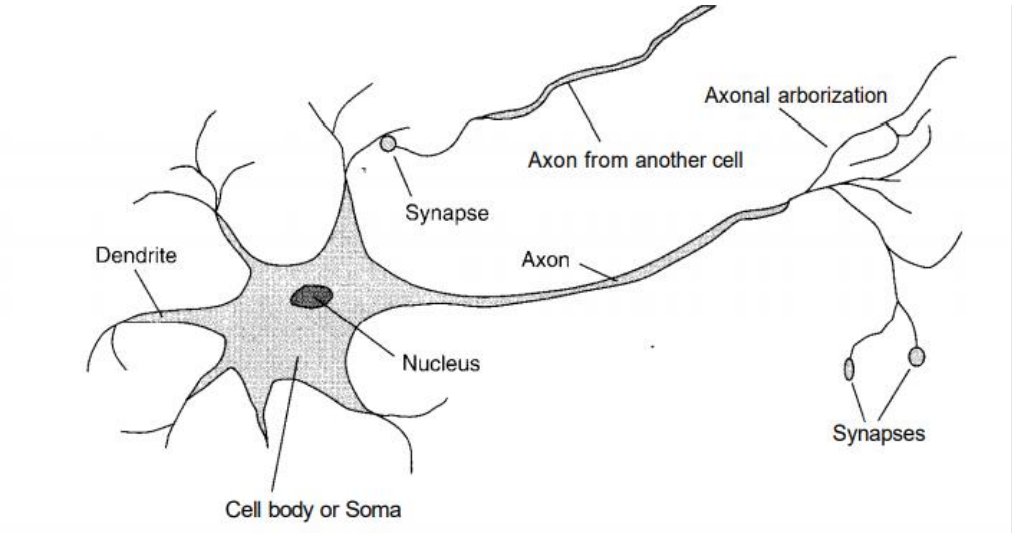


Figure 1 Part of a nerve cell (Russell & Norvig, 1995, fig. 19.1)

A neural network consists of units (nodes), inputs and output. The (1) input and the output are connected by links that have a numeric weight. The (2) weight influences

the strength of input, and multiplying input and weight results in a (3) weighted input (sum). A (4) activation function is applied to the weighted sum, forming the (5) output. [See **Fig. 2**] All these components form a perceptron - a network. The first network introduced was a single-layered feed-forward perceptron [See **Fig. 3 Single Perceptron**] which means the network has many inputs but only one output with no hidden layers where the process is always forward and cannot loop back [See Fig. 2]. A single-layered network has many inputs, but this time it has more than one output. [See **Fig. 3 Perceptron Network**] The perceptron learns through a gradient descent which is one of many weight optimisations options (Russell & Norvig, 1995, Chapter 19.4), but they do not resolve the non-linearly separable problem (Taud & Mas, 2018, Chapter 27).

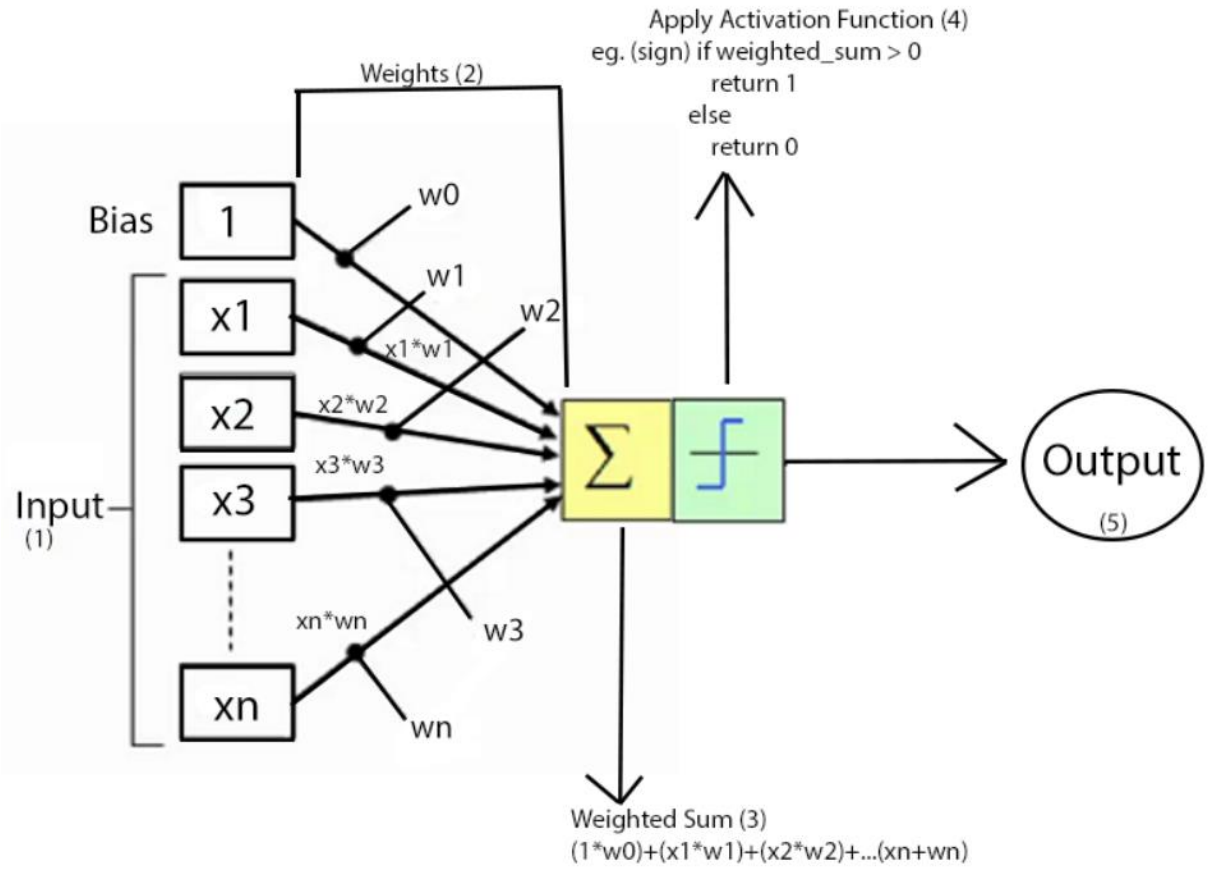


Figure 2 Simplest ANN, Source: (Adapted from Russell & Norvig, 1995, Fig 19.4; Daume III, 2012, Chapter the Perceptron Fig. 3.2)

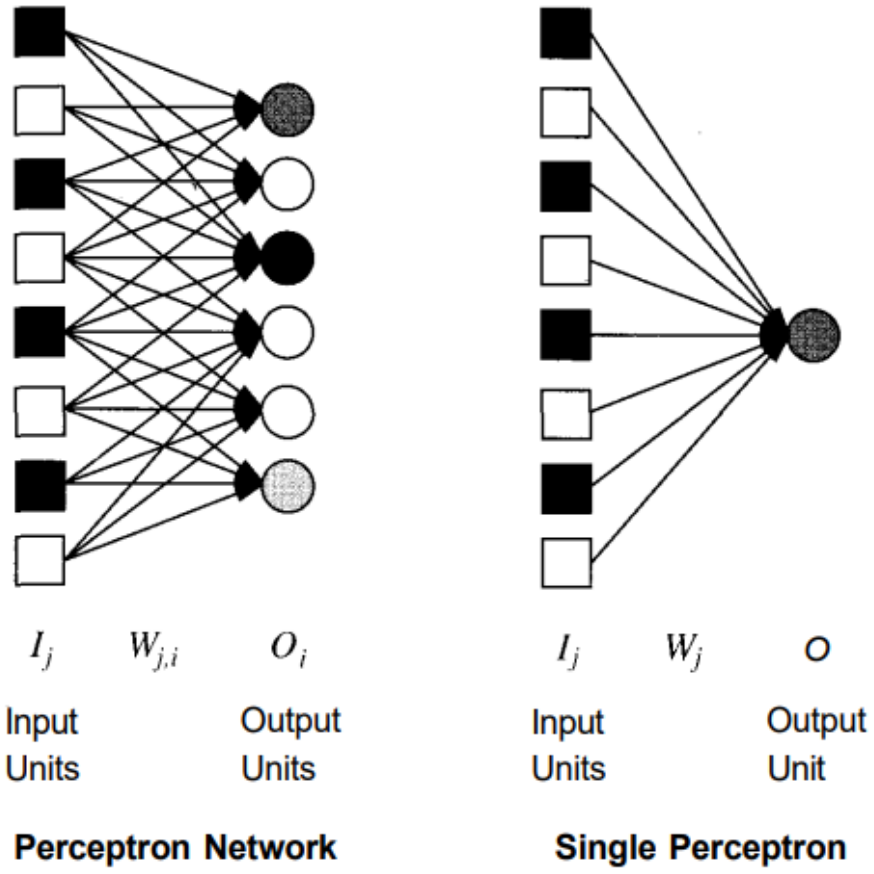


Figure 3 Single Layered Perceptron (Russell & Norvig, 1995, fig. 19.8)

A linearly separable problem means that if we have, for example, some positive and some negative examples, we would be able to see a straight line between the positive and negative, where the positive is on one side and the negative on the other. The perceptron fails if a straight line cannot classify a dataset, meaning the problem is non-linearly separable. The Perceptron and Single-Layered Neural Networks limitations are that the decision boundaries can only be linear and a simple example is that even the XOR problem is non-linearly separable (Hubans, 2020, Chapter Artificial Neural Networks, Daume III, 2012, Chapter The Perceptron).

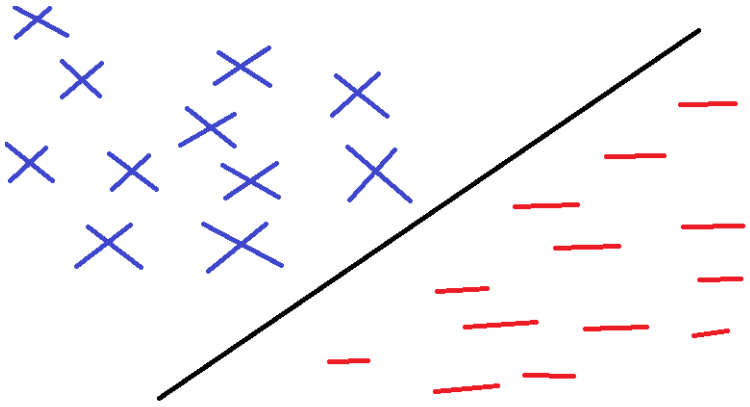


Figure 4 Linearly Separable Problem Source: (Adapted from Daume III, 2012, Fig. 2.9)

2.1.2 Multi-Layered Perceptron

Adding any number of layers in the network to overcome the non-linearly separable challenge forms a Multi-Layered Network [See **Fig. 5**] with the same structure as the single-layered perceptron but has hidden layers between the input and output. A link connects every hidden layer with weights. The only difference between a multi-layered and a single-layered network is that the neurons in the hidden layer (unit) also use an activation function, and every hidden layer could have a different activation function. Referring to **Fig. 4,** the weighted inputs are summed in the hidden layer, and an activation function is applied, producing the output.

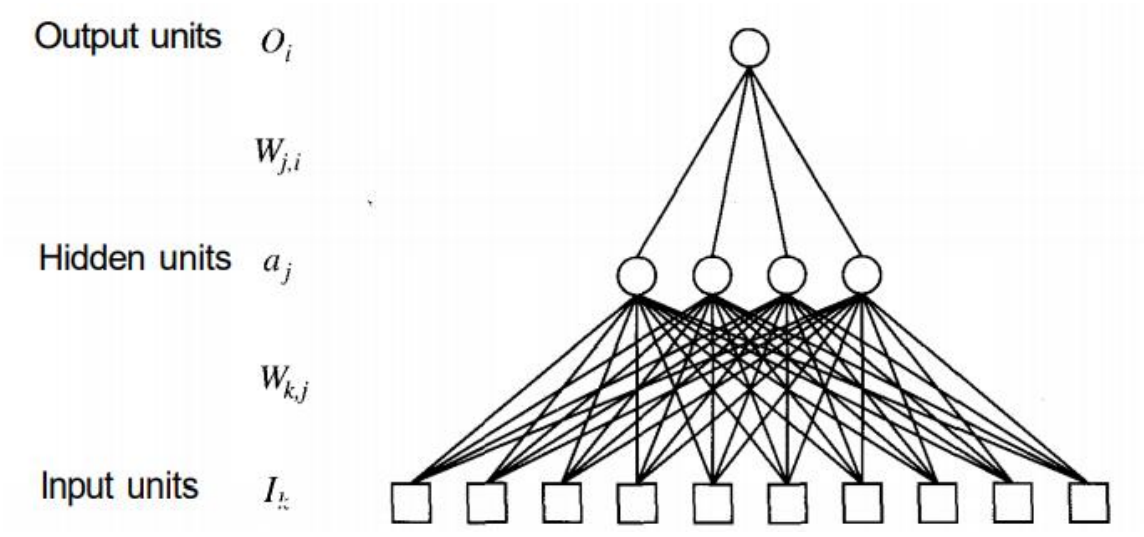


Figure 5 A two-layered Feed-Forward Network (Russell & Norvig, 1995, fig. 19.13)

In a multi-layered network, many weights (a weight connecting each input to output) contribute to more than one output, which can be observed between the input units and the hidden units in **Fig. 5**. (Russell & Norvig, 1995, Chapter 19.4). After applying

the activation function to get the output, which is the last step of the forward-propagation, the next phase is the training phase. In the training phase, the backpropagation algorithm helps the model learn from its mistakes by calculating the cost difference between the predicted output and the actual class for the examples in the training set. When the cost difference is calculated, the weights are updated to be smaller or larger accordingly, improving the network. We divide our dataset into a training set and a test set. The training set consists of many different examples that we use to train our neural network over several iterations, where on each iteration, the weights are adjusted. (Daume III, 2012, Chapter Neural Networks, Hubans, 2020, Chapter 9. ANN) After we teach the network, we use the test set to see how well our neural network performs on a given task.

2.1.3 Learning Methods

The primary learning methods in neural networks are supervised, unsupervised, and reinforcement learning [See **Table 1**]. According to Russell & Norvig (1995, Chapter 2), an agent is considered anything that can perceive its environment through sensors and acts upon that environment through effectors. [See **Fig. 5**]. This project focuses on artificial neural networks used for control where the output would not fall under a class. Is this a human or an animal? However, in this project, the output would be an action for our agent to do (an action), should the agent go left or right, and as input, it takes from its sensors.

In contrast, in supervised learning, usually the output from classification usage predicts the category (class) of an example. To increase its accuracy, we input many labelled instances of a specific class as we select features that the neural network uses to update its weights to detect that that is indeed an animal or a human. Supervised learning was the primary method used to control a robot arm in a neural net to learn to control where correct behaviours are explicitly given (input and output), where training examples that demonstrate the correct mapping from input to output is required. As training examples, recorded joint movements and final arm positions were generated by swinging the arm. However, that makes it difficult to avoid obstacles because of the difficulty of generating examples of obstacle avoidance, so Moriarty and Miikkulainen (1996a) present an alternative approach that evolves neural networks through genetic algorithms a neural network without explicitly providing training examples but instead

learn through evolution. Finally, the evolution is guided through a fitness evaluation, and rewards and penalties help the neural network learn.

In an agent's case, the classic example of control is a vacuum robot supposed to clean a dirty floor. The vacuum cleaning robot is a control task that takes information (input) from the robots' sensors, like a camera, which is its precepts. The output is not an action. What speed should the motors run at or from the vacuum cleaner, move left, proper or clean, so it is essentially controlling the robot's behaviour? The robot adjusts to please the fitness evaluation through the entire task, involving multiple skills. In this paper, we would be referring to the same kind of task, but rather than a physical robot, we would control a virtual agent.

Category	Supervised		Unsupervised		Reinforcement
Input and Output	Feed network with labelled input and output (encoded as pairs)		Feed network with labelled input with no output data		Agent interacting with the environment [do not have access to correct outputs that are connected to the input]
Usage	Classification	Regression	Clustering	Anomaly Detection	Games, Robots
AIM	Predict categories of examples based on features (class)	Predict continuous values	Grouping similar objects together	Identification of rare items or events	Train an agent in an environment based on rewards and penalties
Overview	Learning from examples		Discovering underlying patterns that we may not see		Learning behaviour by interacting in an environment and receiving rewards (positive or negative)

Table 1 Types of Learning, Source: (Adapted from Russell & Norvig, 1995, Chapter 18; Russell & Norvig, 1995, Chapter 20; Brown, 2014, pt. Neural Networks for Prediction)

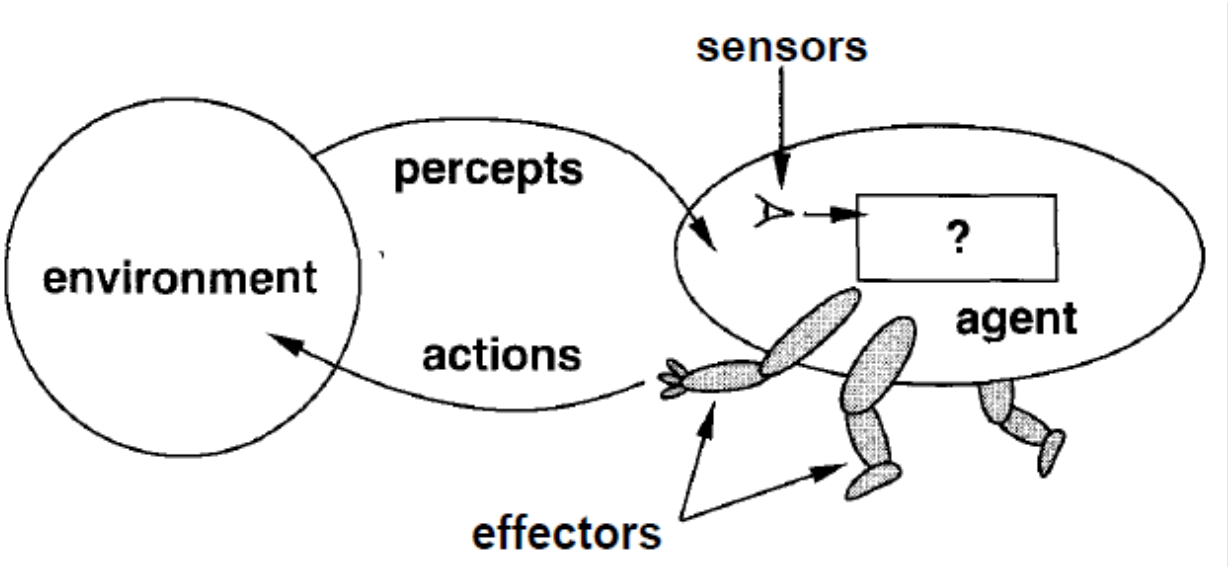


Figure 6 Agent interacts with Environments through sensors and effectors (Russell & Norvig, 1995, fig. 2.1)

2.2 Neuroevolutionary

Neuroevolution is a technique that adjusts neural network weights or topologies to learn a particular task where explicit training examples are not required. Evolutionary computation is used to search for parameters in a network that increase a fitness function that evaluates execution in a task. Neuroevolution is highly general, allowing learning without explicit targets, nondifferentiable activation functions and recurrent networks (Stanley & Miikkulainen, 2002b). Neuroevolution has been used successfully on various tasks, such as robot control, music generation, modelling biological phenomena, and chip resource allocation. The reason that it has broad applicability is that it can be cast as an on-optimisation problem. Different NE approaches depend on the input representation as different evolutionary algorithms that support different networks (Risi & Togelius, 2015). Stanley and Miikkulainen (2002a, sec. Introduction) states according to past papers on single-pole balancing (Moriarty & Mikkulainen, 1996b), using the SANE method, which evolves individual neurons, proves to be faster and more efficient than some reinforcement learning methods like Q-Learning, and robot arm control (Moriarty & Miikkulainen, 1996a), where an alternative to using classification for control is tested against NE, NE has shown to be faster and more efficient than reinforcement learning methods

2.2.1 Terminology

The Terminology used is shown in **Table 2** for Evolutionary algorithms and for NEAT.

Term	Meaning
Allele	The value of a specific gene in a chromosome
Chromosome	A collection of genes that represent a possible solution
Individual	A single chromosome in a population
Population	A collection of individuals
Genotype	Representation of a solution
Phenotype	Represents a unique solution
Generation	A single iteration of the algorithm
Exploration	The process of finding a variety of possible solutions (some good/some bad)
Exploitation	Process of honing in on reasonable solutions and iteratively refining them
Fitness function	A particular type of objective function
Objective function	A function that attempts to maximise or minimise
Variety	Individuals have different genetic traits
Hereditary	A child inherits genetic properties from its parents
Selection	A tool that measures the fitness of individuals (survival of the fittest)
Survival of the fittest	Stronger individuals have the highest likelihood of survival
Reproduction	two individuals in the population reproduce to create offspring.
Crossover	The offspring created through reproduction contains a mix of their parent's genes
Mutation	The offspring created through reproduction have slight random changes in their genetic code.

Table 2 Terminology, Source: (Adapted from Hubans, 2020, Chapter 6. Advanced Evolutionary Approaches)

2.2.2 Evolutionary Artificial Neural Networks (EANN's)

Evolutionary artificial neural networks (EANN's) belong to a particular class of artificial neural networks (ANN's), where evolution is a fundamental form of adaptation in addition to learning (Xin Yao, 1999). Sapp (2003, Chapter Evolution and Revolution) states that evolution is a confirmed fact from biologists, and the influence from evolutionary thinking has extended beyond biology, for example, in economics,

artificial intelligence, medicine, psychology and more. Darwin's theory changed the approach and understanding of natural history in the nineteenth century, and that monumental change occurred when a comparison was made with traditional Judeo-Christian beliefs and Darwinian theory. In short, the Judeo-Christian belief is that only six biblical days have created everything and formed all species just as they are today with no genealogical relationship between them, wherein Darwin theory all life is related to the genealogical relationship between species. According to Darwinian evolutionary theory, organisms evolve in a makeshift concerning the changing conditions surrounding them, meaning that new organs, for example, do not just appear but have been created for a purpose. The not suitable for the environment organisms die, whereas the fit organisms live to reproduce. Like their ancestors (parents), each new generation of children (offspring) is similar to the previous generation. Evolution takes time and does not happen suddenly, so a rapid change could wipe out species as members evolve with the slowly changing environment. (Russell & Norvig, 1995, Chapter 20.8) Inspired by nature, evolutionary neural networks have a robust way of evolving successful organisms where if the environment changes slowly, the species evolve along with it by a mutation of some kind (Russell & Norvig, 1995, Chapter 20.8). EANN's use of evolutionary algorithms is a fascinating alternative compared to learning algorithms such as backpropagation, as Floreano, Durr, Mattiussi (2008, p. 57) mentioned and according to the research of Whittaker (2012) have proven to be powerful.

2.2.3 Evolutionary Algorithms

The Evolutionary Algorithms are population-based that simulate biological systems' natural evolution where information is exchanged between individuals in a population (See **Fig. 6**). The leading operators are selection, crossover, and mutation, inspired by the Darwinian evolution theory by natural selection. Based on the theory, when reproduction occurs between two individuals, the offspring, due to crossover, inherits genetic properties, and because of mutation, it has a slight change. (Hubans, 2020, Chapter 4. Evolutionary Algorithms).

According to Omelianenko (2019, Chapter Mutation operator), the mutation operator plays an essential role in preserving the population's genetic diversity during evolution. Regarding the mutation probability that is defined, the mutation alters one or more genes in the chromosome. The crossover operator, which means recombination,

generates new generations (solutions) from existing populations by re-combining genetic information from two parents to generate offspring. The crossover operator is helpful because re-combining reasonable solutions from the parent organism could lead to potentially better-combined offspring (Omelianenko, 2019, Chapter Crossover operator). It is the phase where it generates genetic diversity and promotes proper chromosome segregation. (Lorenz & Whitby, 2006)

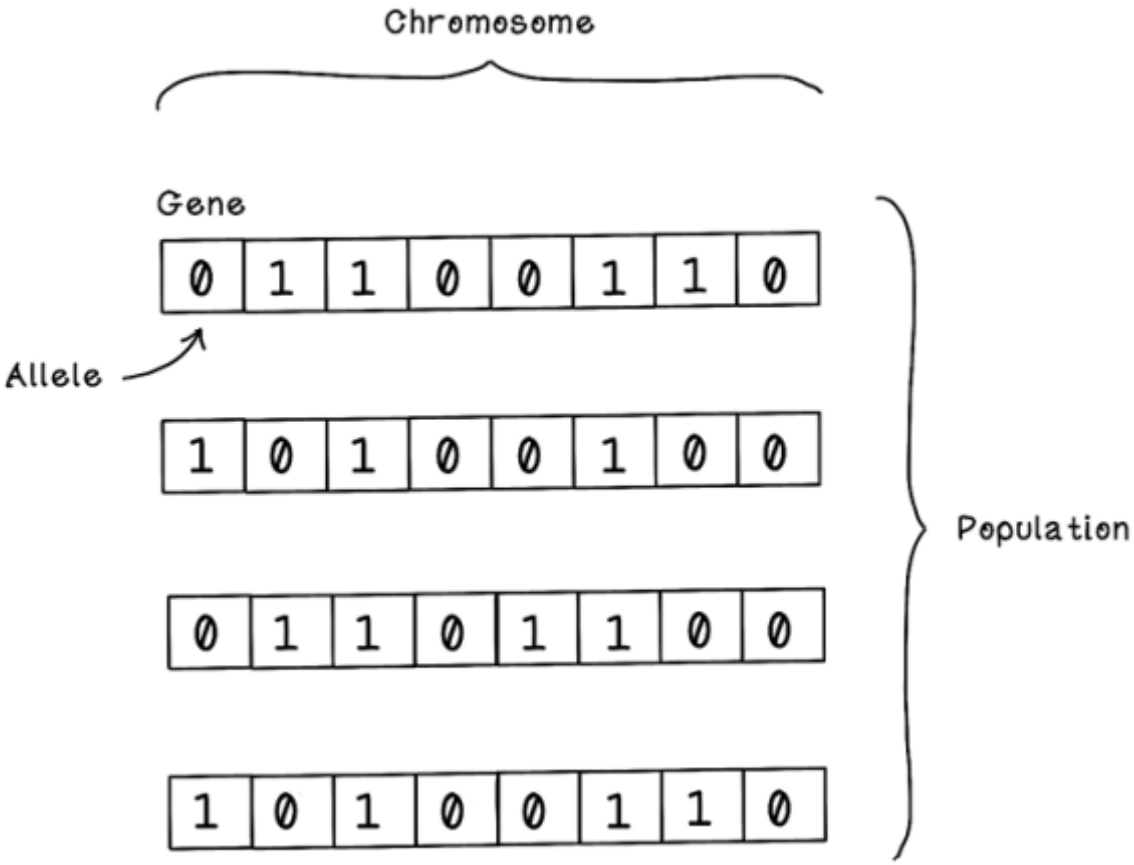


Figure 7 A population (Hubans, 2020, fig. 4.11)

Compared to other learning methods, features defining the neural network can be encoded genetically and co-evolved simultaneously, leading to a more flexible performance criterion. (Ding, Li, Su, Yu, Jin, 2013)

2.2.4 Genetic Algorithm

According to Ding et al. (2013), the Genetic Algorithm (GA) is good at a global search compared to other algorithms. What makes the genetic algorithm reliable, as stated

by Mirjalili (2018, p. 54), is the process of maintaining the best solutions in each generation and using them to improve other solutions as the entire population becomes better each generation. Another advantage is the mutation that randomly changes the chromosomes' genes to maintain the population's diversity.

2.2.5 Comparison with Backprogatation

Compared with Backpropagation (BP), both algorithms are sensitive to the parameters, but while BP is used more effectively in a local search, it cannot handle discontinuous node transfer functions. GA is successful in searching globally. (Ding et al., 2013).

Örkcü & Bal (2011) point out the differences between GA and BP:

- GA considers many points in the search simultaneously,
- GA works with strings of characters representing the parameter set, not the parameters themselves as in the BP algorithm.
- GA does not use deterministic rules. It uses probabilistic rules to guide.

They conclude that the real-coded genetic algorithm can provide good classification results as well, just like backpropagation. In another paper by Chukwuchekwa (2011), he investigates what happens when BP and GP are used in a feed-forward neural network for pattern recognition. They find that the BP training algorithm is more successful than the GA, but the GA's performance indicates that It can be used as an alternative training algorithm.

2.3 Atari Games

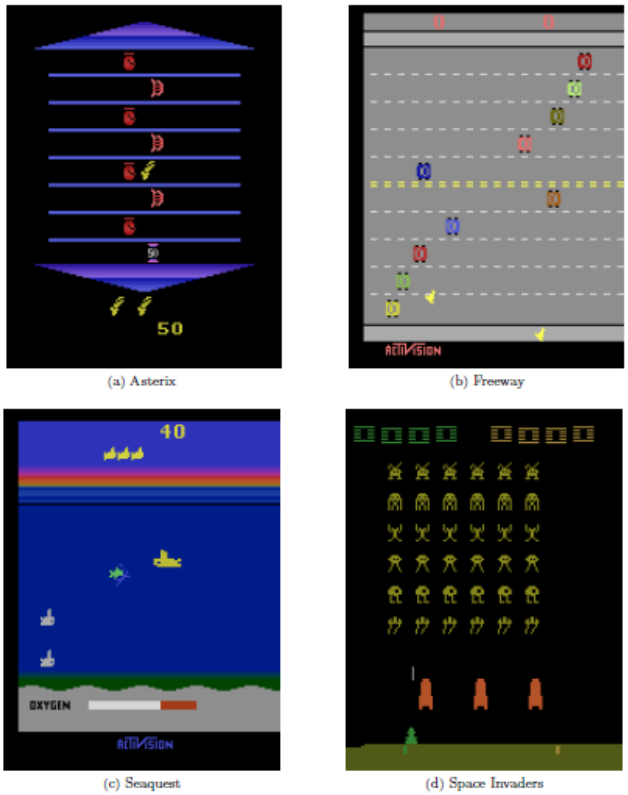
2.3.1 What are the Atari Games?

Atari 2600 games vary between classic board games and graphically intensive video games that include several genres such as action-adventure, board games and shooting games. Regarding AI, these games, due to the 2D game screen and joystick manipulation (state and action interface), makes it an excellent environment for researchers as it is much like modern video games but provide the agent with a simple enough interface but complex enough dynamics that are interesting [See **Fig. 7**]. The state can be described simply by 2D graphics that contain between 8 and 256 colours and the resolution being 160 pixels wide and 210 pixels high. The discrete action space

consists of eight movement directions for the joystick with an extra single button (Hausknecht, Lehman, Mikkulainen, Stone, 2014).

2.3.2 What research has been done?

Hausknecht et al. paper (2014) stated that the Atari domain was first used as a research platform with Pitfall (Diuk, Cohen, Littman, 2008). Commands from the joystick were sent to the Atari emulator while introducing an object-oriented representation for reinforcement learning problems for factored-state representation with stochastic transitions. Simple heuristics were used to identify objects by their colour cluster. Then it was used as an interface with learning agents (Naddaf, 2010) where AI agents are developed without any game-specific prior knowledge or assumptions using reinforcement learning. The agents use feature vectors generated from the game screen as well as the console RAM to learn to play a given game and search-based methods that use the emulator to simulate the consequence of actions into the future, aiming to play as well as possible by only exploring a tiny fraction of the state-space. The software framework Arcade Learning Environment (ALE) used to develop agents that play the Atari games quickly was afterwards released (Bellemare, Veness, Bowling, 2013, sec. 2). The ALE has been used as a domain for reinforcement learning and search agents, where Bellemare et al.(2013) introduces the ALE, where he illustrates the promise by developing and benchmarking domain-independent agents using AI techniques for reinforcement learning and planning. The Atari games were also used to investigate aspects of prediction and self-detection, recognising some future observations under the agent's control. In contrast, others are entirely decided by the environment (Bellemare, Veness, Bowling, 2012), where he finds that contingency awareness can significantly improve the performance of existing feature construction methods. This paper uses the Atari Games emulator to build our neural networks because we can see from the already performed research. However, the Atari games as quite outdated in comparison with current games. They are valuable for researching because of the straightforward interface, wide variety of games and limited control options.



(a) Asterix (b) Freeway (c) Seaquest (d) Space Invaders

Figure 8 Atari Games (Naddaf, 2010, fig. 1.1)

2.4 NEAT (NeuroEvolution of Augmenting Topologies)

NeuroEvolution of Augmenting Topologies (NEAT) is a method for evolving Artificial Neural Networks with a Genetic algorithm. The NEAT software was designed initially in C++ code intended for the Linux operating system by Kenneth O. Stanley and Risto Miikkulainen (Stanley et al., 2014). In the original paper (Stanley & Miikkulainen, 2002), they present the NEAT method they created to try to answer the question in Neuroevolution on how to gain an advantage from co-evolving neural network topologies with weights. NEAT by design minimises the dimensionality of the search space of connection weights, so there is a significant performance gain if the structure is evolved so that the topologies are minimised and grown incrementally. Traditional incremental evolution of the structures presents technical challenges. In other words, in traditional NE approaches, a topology is chosen for the networks before the beginning of the experiment, where the network usually is a single hidden layer of neurons, but NEAT is designed to address the problems. Due to the topologies being minimised throughout evolution, NEAT has improved efficiency results rather than only the end. (Stanley & Miikkulainen, 2002b).

From Stanley & Miikkulainen (2002b), we can conclude that NEAT evolves both the connection weight and topology employing crossover and two types of structural mutation:

- Add connection mutation – connecting two unconnected nodes with a new connection gene a random weight.
- Add neuron (node) mutation – a new node replaces the old split existing connection.

2.4.1 Advantages of using NEAT

The NEAT method uses Genetic Encoding to solve the issues displayed in the problems of TWEANNs that are shown in **Table 3**.

Encoding	What is it?	Where is the problem?	Problem?
Binary Encoding	Binary encoding represents a gene in terms of 0 or 1, so a chromosome is represented by a string of binary bits.	Size of connectivity matrix - number of nodes^2	Representation issues for a large number of nodes
		The size of the bit string must be the same for all organisms; define the maximum number of nodes initially	If the maximum is not sufficient, the experiment must be repeated
		Using a linear string of bits to represent a graph structure	Makes it difficult to ensure that crossover generates valid combinations
Graph Encoding	Represent graph structures	PDGP uses graph encoding so the subgraph can be swapped in crossover	We cannot be sure whether the subgraphs being combined are the right ones to create a functional offspring
Cellular Encoding	The transformations are motivated by nature in that they specify cell divisions, and genes can be reused during development	Type of Indirect Encoding	Requires more detailed knowledge of genetic and neural mechanisms

Table 3 TWEANN Encoding issues, Source: (Adapted from Stanley & Miikkulainen, 2002b, Chapter 2)

The several specific problems shown in **Table 4** and **Table 5** are derived from **Table 3** and NEAT. Stanley & Miikkulainen (2002b) central intuition behind NEAT comes from representing different structures, which means their representations will not match up. In nature, there is a similar problem with the gene alignment, so nature's solution is to utilise homology (two genes are homologous if they are alleles of the same trait). To solve the competing conventions, which means having more than one way to solve a weight optimisation problem with a neural network, NEAT uses explicit fitness sharing that forces individuals with similar genomes to share their fitness payoff. Therefore the population divides into several species, each on a different peak, without the threat of anyone taking over. By this, the innovations in NEAT are protected in their species using historical information about the genes. NEAT by design starts with a minimal population and grows its structure from there, which solves the problem with the initial population and topological innovation that TWEANN's have [See **Table 5**]. The goal is to minimise the final product and all intermediate networks along the way, so by starting minimally, we reduce the parameters that must be searched, so the system searches for the solution in the lowest-dimensional weight space possible throughout all generations. (Stanley & Miikkulainen, 2002)

Protecting Innovation with Speciation (competing conventions)			
What?	In TWEANNs, innovation takes place by adding a new structure to networks through mutation	[Partly Solution] The GNARL system addresses the problem of protecting innovation by adding a nonfunctional structure.	Speciation (niching) has not been brought in Neuroevolution.
Problem?	Adding a new structure causes the fitness of a	A node is added to a genome without any connections, hoping that some	Networks with structural innovations are not protected, so they

	network to decrease.	valuable connections develop in the future.	have not been given a chance
Why?	It is unlikely that a new node or connection just happens to express a useful function as soon as it is introduced.	Nonfunctional structures may never end up connecting to the functional network, adding extraneous parameters to the search.	it requires a compatibility function to tell whether two genomes should be in the same species
What needs to be done?	Necessary to protect networks, so they have a chance to make use of their structure		

Table 4 Competing Conventions, Source: (Adapted from Stanley & Miikkulainen, 2002b, Chapter 2)

	What?	Problem	Why?	What needs to be done?
Initial Population and Topological Innovation	In many TWEANN systems, the initial population is a collection of random topologies	Random initial populations - a chance that a network has no path from its inputs to its outputs.	Starting with random topologies does not lead to finding minimal solutions	Altering the fitness function in a way to encourage smaller networks

Table 5 Population and Innovation, Source: (Adapted from Stanley & Mikkulaineu,2002b, Chapter 2)

2.4.2 Related projects that use NEAT

2.4.2.1 Game related projects

NEAT has been tested if it would successfully play the games as they are a good test domain, so recent studies (Tupper & Neshatian, 2020) show that surprisingly simple and small neural networks could play the Atari games, effectively separating state representation and policy learning.

The game industry NEAT has been used for various occasions, mainly testbeds due to scripted commercial games, so the computer agent is non-adaptable. NEAT has been tested if it would successfully play the games as they are a good test domain,

so recent studies (Tupper & Neshatian, 2020) show that surprisingly simple and small neural networks could play the Atari games, effectively separating state representation and policy learning. For example, NEAT has been used to make the game more challenging with improving the ghosts' performance strategy in Pacman, where ghosts show teamwork skills using computational intelligence techniques (Wittkamp, Barone, Hingston, 2008). Another example is using NEAT to evolve competitive non-player characters for a racing game where two basic skills are needed: drive fast and reliably on a wide range of tracks; effectively overtake the opponents avoiding the collision. The skills were evolved separately in the experiment and then combined in a single controller, specifically for the Open Racing Car Simulator (TORCS). The experiment was successful when the two evolved neural nets for the skills were combined. Otherwise, the skills alone do not lead to a competitive and reliable car controller. (Cardamone, Loiacono, Lanzi, 2009). Traish & Tulip (2012) perform four experiments on the real-time strategy (RTS) environment Wargus to address if NEAT could generate effective AI agents for real-time strategy games and if NEAT could create agents no single successful strategy having complex and varying behaviour. [See **Table 6**]

Experiment	Intent	Outcome	Conclusion
Agent evolution against a range of single static AI opponents	Can NEAT evolve an RTS AI agent that has basic mechanisms?	Production of effectively integrated agent	NEAT successfully creates an agent that counters a single strategy
Like the above experiment but the agent has to defeat all static AI	Can NEAT evolve agents with complex behaviour?	Although a single strategy agent evolved to defeat all the opponents on each run, successful challenger generation.	NEAT would discover a single simple solution to a complex problem (if any, it could be found) before any adaptive behaviour occurs.
New challengers are required to defeat all	Would NEAT Evolve complex behaviours when	The number of champions grew faster than	Many minor refinements of a simple basic

champions before being added to the pool ("dominance tournament" system)	faxing various opposing strategies?	expected, so execution times were long, resulting in a slow evolution	behaviour were found before any significantly different strategies occurred.
A set of agents, with the task of defeating the whole set of opposing AI, implementing a cycle of counters was forcibly evolved and then used as a static AI set.	Eliminating the possibility of a successful 'degenerate strategy' occurring, thus forcing complex behaviour to evolve.	Demonstrated evolution of complex adaptive behaviour in agents.	Recognises issues from experiment 3 and attempts to evolve adaptive behaviour by preventing a single solution at the cost of computational effort.

Table 6 NEAT in Real-Time Source: (Adapted from Traish & Tulip, 2012)

2.4.2.2 Other Projects

NEAT has been used for experiments like robot control applied in a duel against another robot, where they start on opposite sides facing away from each other, and their goal is to gain a higher level of energy than the opponent (by consuming food). The experiment shows that the genome's complexity leads to the continual coevolution of enhanced complex approaches (Stanley & Miikkulainen, 2004). Another significant usage is in a vehicle warning system that predicts crashes before they have occurred, where NEAT successfully evolves a warning system in a car driving simulation that predicts an estimated time to crash, saving lives in the real world(Stanley, Kohl, Rini, 2005). NEAT has also been used in predicting the price of stock accurately and faster than other methods (Iuhasz, Tirea, Negru, 2012). Liu, Ghanar and Theodoropolulos (Liu, Ghandar, Theodoropoulos, 2020) apply the NEAT algorithm in a Peer to Peer (P2P) lending problem that based on streaming data, a credit evaluation model is updated, whereby regulating the fitness function to

consider performance on classifying diverse classes the implementation can handle class imbalance.

NEAT contributes considerably to GAs because it shows the likelihood of evolving both optimal and complex solutions simultaneously. (Stanley & Miikkulainen, 2002b) NEAT also provides many opportunities to implement and alter the algorithm in distinct ways, varying on the problem we ought to solve. The vital insight is that it is not the definitive structure for the solution that matters but rather the structure of all the in-between solutions along the way of finding the solution. NEAT will not get entrapped even if weights represent a local best in a fitness space because it can include additional structure and add new dimensions, which means it opens new ways to escape.

2.4.3 Comparing different variants of NEAT

While NEAT uses direct encoding, another variant is HyperNeat It evolves an indirect encoding called compositional pattern producing network (CPPN) that defines the weights of an ANN to solve the problem. The CPPN determines the weights between the input layers, where CPPN is said to be geometrically aware (Hausknecht et al., 2014) and can compactly encode patterns with regularities such as symmetry repetition (Risi & Stanley, 2012). Hausknecht et al. (2014) experiment with different kinds of neuroevolutionary methods, and he finds that with object representation, NEAT performs similarly to HyperNEAT, where with noise representations, NEAT outperforms HyperNEAT despite the ability to incorporate geometric information. Noise representation is the resolution downsample of the 160x210 native screen resolution, a sanity check to ensure the neuroevolution algorithm using the object and pixel representation is doing more than memorising a sequence of moves. In the paper, his results indicate that direct-encoding methods work best on compact state representations (low-dimensional, pre-processed object and noise-screen representation).

In contrast, indirect encoding methods allow scaling to higher dimensional representations and learning based on general raw-pixel representation. Another variant of NEAT, which originated from HyperNeat, is Evolvable substrate (ES-) HyperNEAT, representing the encoding pattern of connectivity across a network containing implicit information that expands the scope of neural structures (Risi &

Stanley, 2012). HyperNEAT is particularly well-suited for problems where geometric knowledge is essential, but it loses the advantage as the problem increases complexity. (Stanley, D'Ambrosio, Gauci, 2009, Lowell, Birger, Grabkovsky., 2011). Another NEAT use generates game content like levels, models, textures, and an algorithm content-generating NEAT (cgNEAT). This algorithm automatically generates graphical and game content based on user preferences while playing in real-time (Hastings, Guha, Stanley, 2009). Other NEAT variants include the real-time NEAT(rtNEAT), which makes simulated agents' behaviour more realistic in real-time games. Many Non-Playable Characters (NPC) achieve this high level of realism but achieve a significant time of scripting the behaviour. Reeder, Miguez, Sparks, Georgiopoulos and Anagnostopoulos (2008) use NEAT to evolve the robots' behaviour in NeuroEvolving Robotic Operatives (NERO). However, NEAT alone has some drawbacks in a real-time situation as an entire population of game agents is changing simultaneously, so a human user is likely to notice the difference leading to unrealistic gameplay.

2.5 Summary/Conclusion

This chapter has reviewed Artificial Intelligence's concept and gave a brief overview of what it is, how neural networks work. Furthermore, it looked at the Atari domain and showed crucial reasons why we would use it. Using NEAT has many advantages in evolving our neural network to control a game character, so the aim would be to enable users to do so without writing any code.

Regarding the work done with NEAT, the implementations that use the library in a domain for an experiment need a lot of coding experience is required to perform. This project is essential because it would be implemented in an easy-to-use design without writing any code to train a neural network to play the game. While other machine learning methods have been implemented into software that a user would not need programming experience to use, such as supervised learning tasks, NEAT, and overall control methods have not been implemented into an easy-to-use software. A similar project is accessible in GitHub (Levy, 2020/2021), but the software needs coding knowledge to use it.

This project aims to fill the gap and hence significantly increase the accessibility of NEAT to people who are not comfortable with coding.

3 Requirements Engineering

This section describes the planning and design of the application. The phases followed for developing the software are:

1. Requirements Analysis Phase
2. Designing Phase
3. Coding and Testing Phase
4. Implementation Phase
5. Maintenance Phase

This section discusses considerations regarding the development of the application and what should be implemented. The primary purpose of this section is to identify essential features that the application needs to be easy to use from a user.
The project aims to make the NEAT algorithm more accessible to students and users interested in Artificial intelligence. Hence, the requirements specify how would the goal be accomplished based on testing the NEAT algorithm with various games, different configurations and following the official Neat-Python guidance online (Welcome to NEAT-Python's Documentation! — NEAT-Python 0.92 Documentation, n.d.)
.
The types of requirements are (Saikat Dutt & Geetha Chandramouli Subramanian, 2015, Chapter Introduction):

- Functional Requirements - defines what the system will do or accomplish
- Non-functional Requirements - specifies criteria that can be used to judge the operation of a system, rather than specific behaviours
- Interface Requirements – interaction between other systems

The requirements of the software are the stepping stones to the success of any project. If the software were to be started without understanding why the software is needed (the project's purpose), the final result might not be satisfactory.
In order to understand what the program must do and why the program is needed, the following steps are followed in order to gather the requirements of the system (Saikat Dutt & Geetha Chandramouli Subramanian, 2015, Chapter Requirement Engineering Principles):

- Feasibility Study
- Requirements elicitation
- Requirements analysis
- Specifying requirements
- Validating requirements

3.1 Requirement Development

Before implementing the program, there need to be some requirements that the system must meet to be a successful program. This section has been written following the Requirement Engineering Principles. (Saikat Dutt & Geetha Chandramouli Subramanian, 2015, Chapter Requirement Engineering Principles)

3.2 Feasibility Study

The feasibility study helps in deciding whether or not the proposed system is worthwhile. The feasibility study is split into operational feasibility, technical feasibility and economic feasibility to determine if the system would be required and successful.

3.2.1 Operational feasibility

The operational feasibility checks the usability of the software.
The systems contributions to students studying a computer science degree or other users interested in Artificial Intelligence have been determined to be worthwhile as there are no applications that simplify the use of NEAT combined with the library for games from Open AI – Gym. Students who are just starting to learn about Artificial Intelligence might be interested in the project as an example of using a reinforcement learning algorithm. They can see it in action and understand the difference between supervised, unsupervised and reinforcement learning.
The only application found (Levy, 2020/2021) with similar aims is a command-line application where the user still needs to use the command line to launch the application.
Contacting Kenneth Stanley[1], the creator of the algorithm of NEAT in C++, he expressed that the program is an exciting idea and that the program would be educational to play with it.
Trying to get the algorithm myself to work with the Gym library has been a struggle where it took a while until the libraries' versions were compatible with one another. Furthermore, understanding the configuration's file's usage requires carefully reading through the online guidance to understand what the algorithm can do and how to connect it with the Gym library, especially with the Atari Games.
In addition, trying to fix issues with using the libraries, there are many topics in StackOverflow[2], GitHub[3] and Reddit[4] where people struggle to get Neat and Gym working.

3.2.2 Technical feasibility

The technical feasibility checks if the level of technology required for the software development is available.
Given that the software needs to use Neat in python with Open AI Gym, testing if these combination works have been done before starting the implementation of the project. This check has been passed as the compatibility between the combinations works, given that you have the correct versions of everything.
Another technical check was performed for the graphical user interface (GUI) as the project does not require the user to have any coding experience; all of the features have to be included there. For example, research on a library for a GUI has been conducted and testing the libraries from the Nederkoorn article (2021) has been concluded that the technology required for the software implementation is achievable.

3.2.3 Economic feasibility

The economic feasibility checks whether there is scope for enough return by investing in this software system.
During the development of the system, only open-source and free license software's will be used, so there will be no cost in developing the software or its usage.

---

[1] Kenneth Stanley - Developer of NEAT algorithm originally in C++ - kennethostanley@gmail.com (See conversation in Appendix 4 11.4)
[2] StackOverflow - https://stackoverflow.com/
[3] Github - https://github.com/
[4] Reddit - https://www.reddit.com/

The only cost for the development would be the time it would take to provide a working prototype, so a plan for its development needs to be constructed.

3.3 Requirements Elicitation

In this phase, the source for all the requirements are identified, and then by using these sources, the user's needs and all the possible problem statements are identified. This phase has been iterated throughout the sprints to identify better the problematic areas and the needs that the software needs to implement. The requirements have been split into categories as in sections further below. Issues with the development may occur if:

- the users are not sure regarding the requirements of the software – given that no one has tried Neat with Open AI Gym, gathering requirements for this would create a challenge
- there is a communication gap between developer and users – no communication gaps have been identified
- Requirements are conflicting – no issues have been identified
- The requirements are volatile (change a lot during the elicitation stage) – having the project done with the Agile Methodology, adjusting to changes is easier

The following techniques have been used for the requirements elicitation:

- Prototyping – due to the project being developed in an agile way, every end of the sprint requires to have a working prototype, so the technique was rather useful
- Direct observation – observing the common issue of using NEAT with another library has been the main factor of including some additional features of the program, designing it the way it is and including comprehensive guidance on how to use the application.
- Informal discussion with students regarding NEAT algorithm and GYM library on what would a student look for if they were to use the application

3.3.1 Identifying Stakeholders

Before gathering the requirements for the system, it is vital to know the people who can contribute and help gather the requirements. This part is crucial for developing the application's education model as the targeted audience needs to be identified. The expected points between the groups of users are the focus, motivation (This is why the education mode is designed to be interactive with the user.) and skills (The lessons start progressive starting from the basics of Artificial Intelligence) (Huang & Soman, 2013) (See Appendix 4 11.2). Types of stakeholders:

- People with NEAT Experience
    - People with coding experience
- People with no Neat Experience
    - People with coding experience
    - People with no coding experience

3.3.1.1 People with no Neat Experience

Students studying a computer science degree have been identified as stakeholders as the software should significantly increase their knowledge in Artificial Intelligence. Throughout the development of the program, a few interviews with students have been done to identify what the software would need to be easy to use and understandable. Even though no one I have spoken with knew about the NEAT algorithm, they were interested in learning more about it.

The supervisor of this program is another stakeholder with whom weekly meetings have been occurring to identify the program's progress, any issues the program may have and features that might be useful to the program for it to be easy to use.
Users with no coding experience are the other stakeholders where a demo of the app has been shown often to understand if anything within it would cause frustrations or another approach should be taken for developing the feature or application overall. For example, a considerable amount of time was taken to provide colours that do not cause any issues for the (at least) partly colour blind[5]. Education mode was built in consideration for the people with no coding experience and no knowledge of Artificial Intelligence.

3.3.1.2 People with NEAT Experience

Various forms have been checked throughout the development of the application to understand the most significant issues a user may have with getting NEAT to run. For example, people with NEAT experience should have some form of coding experience. However, the application's normal mode has been developed to make using NEAT less complicated and error-free.

3.4 Functional Requirements

Functional Requirements are the main drivers of the application architecture of a system. They define what the system will do, like react to inputs or accomplish, like train a neural network using NEAT or load the winner. They are features that allow the system to function as intended and are constructed after testing NEAT and the Gym environment.
The functional requirements are split into:

- Education mode
- Editor
- Train Neural Network (Neat Setup)
- Load Winner/Checkpoints

The requirements are prioritized using the MoSCoW Prioritization categories.

3.4.1 Education mode

After careful consideration on what might be helpful for a user in education mode, the functional requirements for education mode have been constructed as follows:

3.4.1.1 Must-Have

1) A user shall be able to navigate easily through the education mode
2) A user shall be able to navigate through the other functionalities while in education mode
3) A user shall be able to choose from a list of lessons
4) The system shall provide easy to follow examples that are working and executable
5) The system shall give To-Do tasks to the user in order to get familiar with the application.

[5] Tested with Plamen Nikolaev Dimitrov – partly color blind user (speaking in person)

3.4.1.2 Should have

6) A user shall not be able to progress into a higher up lesson if a previous one has not been viewed
7) A user shall be able to save the progress of education mode and reset it
8) The system shall load the content of the education mode fast after it has preloaded the content (so it is not loaded every time each lesson is opened)

3.4.1.3 Could have

9) The system shall indicate with a scroll bar if there are more lessons
10) A user shall be able to switch between education mode and normal mode (hide education mode)

The whole education mode is focused on people who have no experience with NEAT and Artificial Intelligence, and it is developed to further their knowledge, allowing the user to understand basic concepts and run Neat.

3.4.2 Editor

In order to have total freedom to edit the configuration file and be easy for a user to use it successfully, the following functional requirements have been made:

3.4.2.1 Must-Have

1) The system shall include all of the parameters of the config file and validate the values imputed.
    A. furthermore, the system shall provide validation for the values that are written into the text fields, so the user knows if what they are inputting is correct
    B. furthermore, the system shall detect if nothing or wrong input is imputed
        a) the application should indicate that either the value is wrong and make the field empty or insert the default value if the parameter has one
2) The application shall include an easy way to transfer the values from the forms onto the config file (config file and values should always remain visible)
3) The application shall include an option to save the text (config) that has been edited.
4) The system shall be able to detect empty values of parameters and not overwrite parameters that have values in Editor
5) The system shall be able to add any parameters with values from the form when they are not included in the config file loaded in Editor

3.4.2.2 Should have

6) The editor shall include a text field where the user can load:
    A. an existing config file
    B. an empty config file (with parameters where their values are empty)
    C. a config file with the default values for the parameters
7) The editor form shall include an automatic function where the user would not need to manually insert the input nodes and the output nodes as they are dependent on the game

3.4.2.3 Could have

8) The system shall include a description of the parameter and its proper name in a static pop-up message that will not interfere with user input – for people with little or no experience in NEAT

3.4.3 Train Neural Network (Neat Setup)

3.4.3.1 Must-Have

1) The system shall include all options that the NEAT algorithm can give a user when training
    A. a user shall be able to select a game from a list of tested games
    B. a user shall be able to specify the winner file name
    C. a user shall be able to know how many checkpoints does the user want to have
    D. a user shall be able to choose from the supported network types
    E. a user shall be able to choose if the game should be rendered (show the window where the neural network is trained) or hide it
    F. a user shall be able to choose a configuration file
        a) from a different folder on the computer
        b) from a text editor (save the text as config file)
        c) restore from checkpoint
    G. the system shall include an extra option for no_fitness_termination = True - the user shall be able to specify for how many generations does he/she want to run the algorithm for
    H. the system will detect if a game is a Box2D game (includes Episodes):
        a) the app will include an option where the user would like to specify how many episodes/tries should a genome run
2) The system will prevent execute the NEAT algorithm if any invalid input is given
3) The system will prevent any games outside of the list to be executed

3.4.3.2 Should have

4) The system will show the output of the NEAT algorithm while it is being executed on a field in the application

3.4.3.3 Could have

5) A user shall be able to edit input and output values depending on the game chosen automatically in the config file
    A. The user will be able to see what changes will be made in the config file

3.4.4 Load Winner

In order to make it easy for a user to use and include all of NEAT's functionality, the following requirements have to be included in the application:

3.4.4.1 Must-Have

1) A user will be able to load all the checkpoints quickly and delete the ones that the user would not like to see
2) The user will be able to specify how many genomes would he/she like to view from the checkpoint (particular generation)

3) The user shall be able to choose a configuration file
    A. from a different folder on the computer
    B. from a text editor (save the text as config file)
4) The user shall be able to see what is being executed on the screen (a winner or a specific checkpoint)

3.4.4.2 Should have

5) A user shall be able to select the winner file from different directories
6) A user will be able to select how many times the winner genome will play the game

3.4.4.3 Could have

7) The system will copy over any values from Run NEAT (Train Neural Network) tab to the Load Winner Tab
8) The system will keep a record of what has been run with what values

3.5 User stories

The functional requirements have been converted into user stories to express the needs of the user in order to use the application successfully.

3.5.1 Education mode

1) As a user, I want to hide the education mode when I do not need it
2) As a user, I want to be able to view the lessons available
3) As a user, I want a guided education that starts from the primary content and progresses.
4) As a user, I want to be able to see working examples on how to execute Run Neat and Load winner/Checkpoints
5) As a user, I want to be able to understand how the neural network works.
6) As a user, I want to save my progress in education mode so I do not have to start from scratch the next time I open the application.
7) As a user, I want to be able to reset the progress in the education mode.
8) As a user, I want the education mode to be a pleasant experience without any loading delays
9) As a user, I want to do some tasks myself to perform the training on a neural network and load a neural network on my own.
10) As a user, I want to see if there are more lessons in the list than the ones displayed.
11) As a user, I want to still navigate through the application for its other functionalities while in Education mode.

3.5.2 Editor

1) As a user, I want to be able to load an existing config file and edit it in the application
2) As a user, I want to be able to save the config file that I have edited
3) As a user, I want to be able to get the default values automatically when loading an empty config
4) As a user, I want to be able to get an empty config file

5) As a user, I want to be able to view all of the parameters that can be included in the configuration file
6) As a user, I want to be able to see the description of the parameter, so I do not have to refer to the guidance online
7) As a user, I want to be able to see if I have written a wrong value
8) As a user, I want to be able to see the default value of a parameter if I leave the parameter empty or input an incorrect value
9) As a user, I want to easily update the config file with the values I have inserted from the form and not have empty parameters inserted.
10) As a user, I want to be able to select a game of my choice from a list of tested games and see get their input and output values onto the form automatically
11) As a user, I want to add any additional parameters that the config file in the editor is missing with the same button that updates the form.

3.5.3 Train Neural Network (Neat Setup)

1) As a user, I want to be able to understand clearly what is needed to train a neural network to play a game
2) As a user, I want to be able to modify any option that can be chosen for the NEAT algorithm (for example: network type, game, configuration file, winner file name)
3) As a user, I want to have the option to choose whether I want to see the neural network is trained in the game or run the training in the background.
4) As a user, I want to see the progress of the NEAT algorithm.
5) As a user, I want to modify the input and output values automatically in the config file after selecting a game, so I do not have to open the config file each time.
6) As a user, I want to be able to write the winner file name and have no issues when saving and loading (name-wise)

3.5.4 Load Winner

1) As a user, I want to see what parameters are needed to load the winner.
2) As a user, I want to get the values that have been used to train a neural network automatically transferred to Load winner, so I do not have to type them again.
3) As a user, I want to be able to see what have I run in the past, and with what values so I can rerun it in the future
4) As a user, I want to choose how many times I would like to test the winner genome per execution.
5) As a user, I want to view all the checkpoints available and delete the ones I do not want to view.
6) As a user, I want to choose how many genomes I would like to see in the checkpoint (generation), so it does not run the whole population.
7) As a user, I want to view the checkpoints as a progress of the training of a generation so I can see the difference between the generations.
8) As a user, I want to view what is being executed on the screen (a winner or a particular checkpoint).
9) As a user, I want to load the winner file from a directory or type in the name in case it is in the program's root directory.

3.6 Non-functional Requirements

The non-functional requirements support the functional requirements. They are used for the operational aspects of the architecture. These requirements may affect the overall architecture of a system rather than an individual component. Each non-functional requirement may generate several related functional requirements
There are three types of non-functional requirements:

- Product requirements – all functions, features and behaviour
- Organizational requirements
- External requirements

3.6.1 Product requirements

- **Usability requirements** - where the interface should be easy to use and understand. The interface of the system is easy to understand and use without further training.
- **The efficiency of use** is easy to accomplish quickly and with few or no user errors. It should not take up much space as employees must use it while running other programs. This includes performance and space requirements.
- **Robustness**: The program must respond to errors where the program crashes and should not restart. With that in mind, users can save their work frequently if the program crashes, so they do not lose progress.
- **Space**: The program must not take too much space on the computer.
    - Currently, there is no executable for the program, but running the program as an executable file creates space issues as each time it is running, it creates a folder in the Temp folder in the C drive which can fill up relatively quickly if the program is run by .exe file multiple times and temp files are not cleared.
- **Portability**: The program can be used on different operating systems.
    - The program currently is only run with the repository from An integrated development environment (IDE)[6]

3.6.2 Organizational requirements

The Organizational requirements are environmental, operational and development that is of importance to organizational policies and procedures. They do not affect the program directly but must be implemented.
The distribution to users is currently via GitHub that includes the whole code of the program and guidance on how to get the program started. Currently, there is no executable file that can be run, but a user would instead need to install some pre-requirements, clone the repository, set up the Virtual Environment and install the dependencies within the program. The creation of the executable made some issues as the Atari Games did not launch as the Visual Studio C++ Build Tools needs to be installed on a machine, so further testing on how to map this dependency is needed to include the program as an EXE file for an operating system.
There are no organizational policies or procedures that a user would need to use the program as it will be open-source. Furthermore, no authentication is needed for the program to be used except logging into GitHub so a user can clone or download the repository.

[6] An integrated development environment (IDE) - a software application that provides comprehensive facilities to computer programmers for software development

3.6.3 External Requirements

The External Requirements arise from influences that are exterior to the system and its development. They include regulatory, ethical, legislative (account requirements and safety/security requirements).

- Ethical – there are no ethical requirements for the program; any user interested in the program can use it.
- Legislative
    - Privacy – the program does not keep any information of the user and does not send reports
    - Safety – the program is safe to use; there is no malware code included in the program. The dependencies have been updated so there are no security vulnerabilities as older versions of some of the dependencies have security vulnerabilities, so the requirements.txt file includes up to date dependencies with no security vulnerabilities found.

3.7 Requirements Analysis

This phase is where the requirements need to be understood in detail as this is the gap between the requirements engineering and the system's design.

3.7.1 System Models

Designing the application considers all of the functionalities that the Neat algorithm should have implemented into the system and other features that would be useful to implement. For example, they were split into different categories and put into different tabs (See Fig. 9 – Access different functionalities). The Text Field is integrated to be always viewable in order for the user to be able to see the config file at all times so the user can understand from (See Fig. 9 – Text Field for Editor):

- Education mode - how the configuration file is used (See Fig. 10 – Features of education mode do not look like sketch due to issues from Tkinter Library)
- Neat Config – see the changes made from the Neat Form to Text Field
- Neat Setup – be able to see the configuration file while the agent is trained or work on the configuration file in parallel while the algorithm is running.
- Load Winner – be able to see the configuration file while you view the winner agent that has reached the threshold or work on the configuration file in parallel while the game is rendered.

The Buttons are integrated where they are in order to be always viewable for the user, so while the user is using the tabs: the configuration file can always be loaded, the colour of the app can be changed, and the mode of the app can be changed (See Fig. 9 – Buttons).

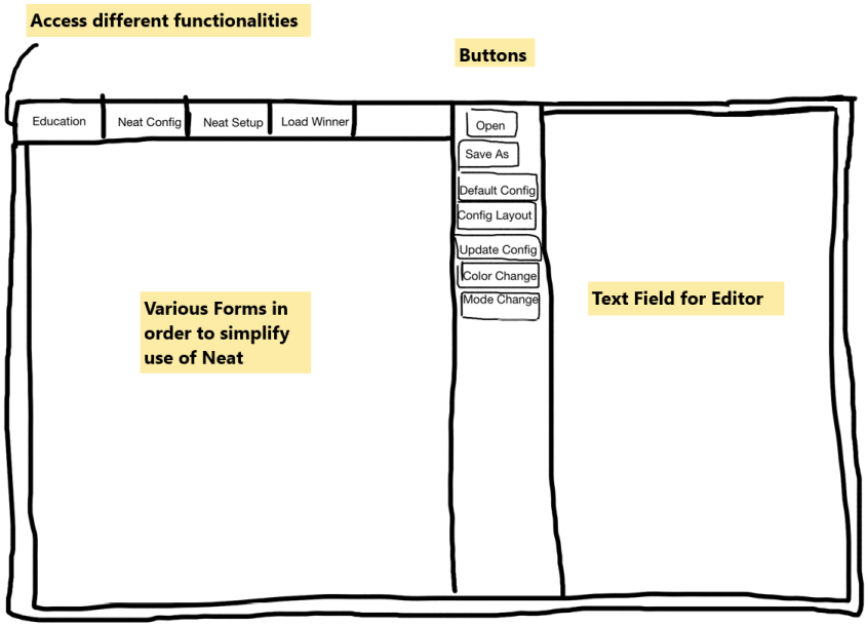


Figure 9 System graphical user interface

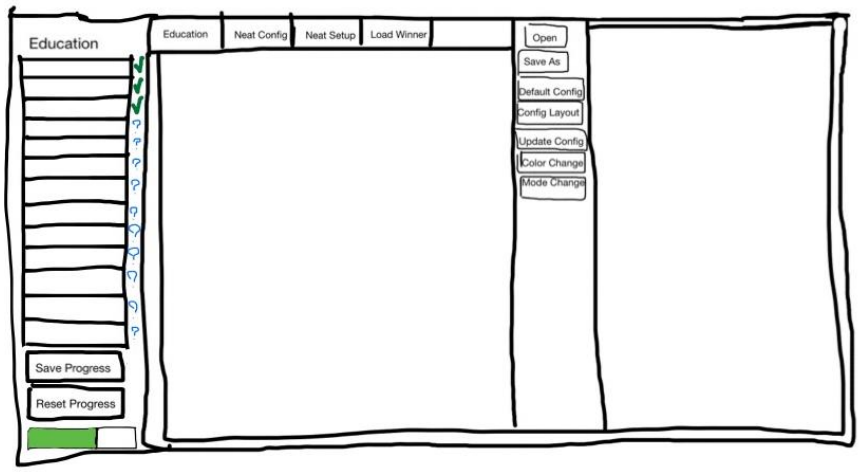


Figure 10 System Graphical User Interface Education Mode

Choosing this specific user interface was from observing many other easy to use applications for a task such as:

- Tableau/Weka – how to make it easier for the user to work with data
- Lobe AI – simple design inspiration for a complicated task
- Any application that has a text form (Weka) – due to NEAT having many parameters and a few modifiable parameters before running and viewing a simple way of displaying them is using a text form
- Game Applications - Helper messages were taken from the game design where a user has to click on an object or hover with the cursor to see what the object is for or what it is.

Many colours were considered for the normal mode, specifically for education mode, such as (Admin, n.d.; Rocheleau, 2020; Pintillie, 2010):

- Monochrome: It is all the same to me
- Opposites: Do they attract?
- Analogous colours: The colour next door
- Triadic: They come in threes

The colours chosen for the development of the application is in honour of the Edinburgh Napier University[7] colour palette for the home page and Moodle Page. Using a web browser and inspector mode, the hex colour numbers were taken from there and integrated the closes one from Tkinter (wpjohn, 2017) into the program.

3.7.2 Use Case Models

When a user launches the application, Education mode and Normal Mode are presented as options. Choosing an option will prompt the application to launch a different mode of the application. (See Fig. 10) The goal for this use case is 1) user story from Education mode where the user can either launch the application in education mode which extends the standard mode – it unlocks the Education tab and unhides the additional frame where the lessons of the educational mode are stored. With this use case, the application is split for users who do not have experience in coding or Artificial Intelligence (Education Mode) and for users who have experience and could have an idea on how to use the application.

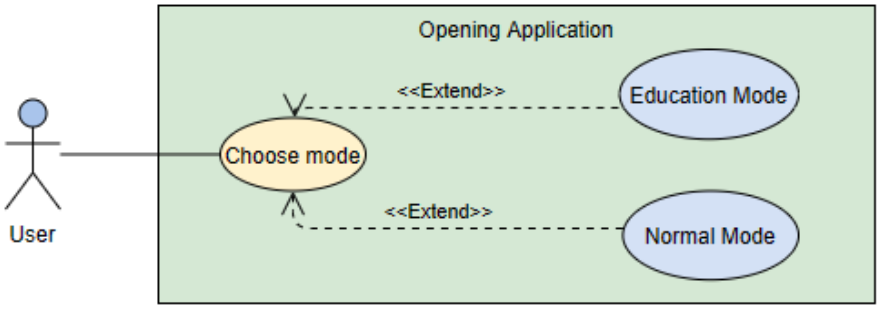


Figure 11 Opening Application

Once the system is launched, the user will see tabs: Education, Neat Config, Neat Setup, Load Winner; These tabs provide the user with the functionality to choose any of the tabs that contain different functionalities, which satisfies 11) user stories from the Education mode (See Fig. 12):

- Education (See Fig. 17)– go through the education lessons displayed in the Education list one by one to learn more about artificial intelligence and how to use NEAT with Gym. The user can save the progress of the lessons via the "Save Progress" button and reset it with the "Reset Progress" button. The use case shown in the diagram is developed following 2) and 6) user stories from Education mode in mind.
- Neat Config (See Fig. 14) – The user stories 5), 6) and 7) from Editor where the user can edit the configuration file within the application and see if the input is wrong while being able to see the description of the parameters
- Neat Setup (See Fig. 15) – Be able to choose setup for the Neat algorithm before training the neural network and launch the training of the neural network (User Stories 1) and 5) from Neat Setup)
- Load Winner (See Fig. 16) – Be able to see the outcome of the training (User Story 1) from Load Winner)

User Stories 1), 2), 3), 4), 9) and 11) (from Editor) are implemented in the Config File buttons, which include (See Fig. 12, Fig. 13, Fig. 15, Fig. 17):

- Open – Open an existing configuration file either to view while executing the NEAT algorithm or edit in Neat Config (User Story 1))
- Save As – save the configuration file from Text Field (User Story 2))

[7] Edinburgh Napier University - https://www.napier.ac.uk/

- Default Config – get an example Config file with the default values for parameters that have (default values for parameters are obtained from NEAT-Python 0.92 Documentation (Welcome to NEAT-Python's Documentation! — NEAT-Python 0.92 Documentation, n.d.)) (User Story 3))
- Config Layout – get an example Config file with empty values for parameters (User story 4))
- Update Config – transfer values from Neat Config into Text Field (User Stories 9) and 11))

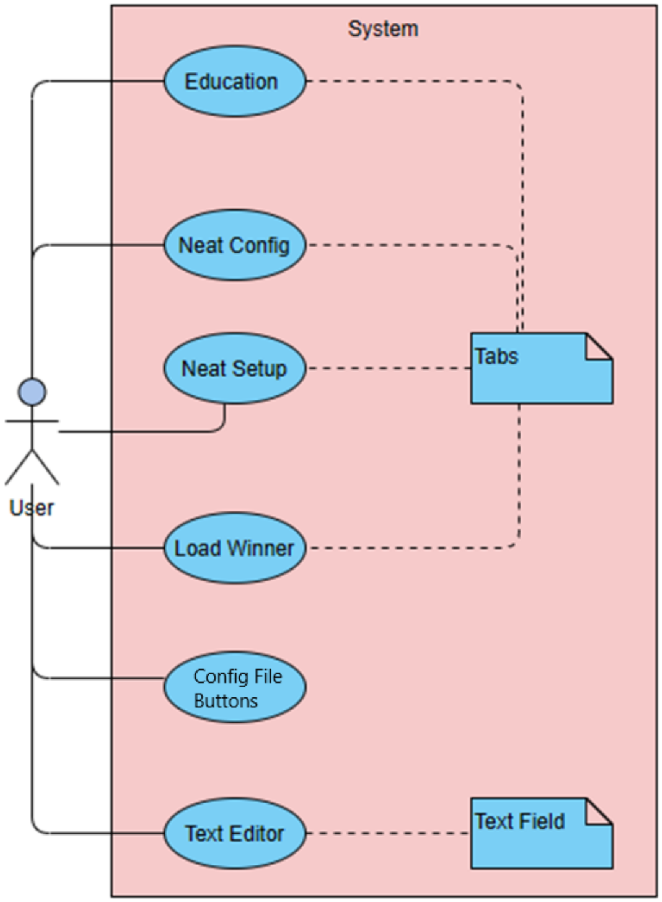


Figure 12 System functionalities Use Case

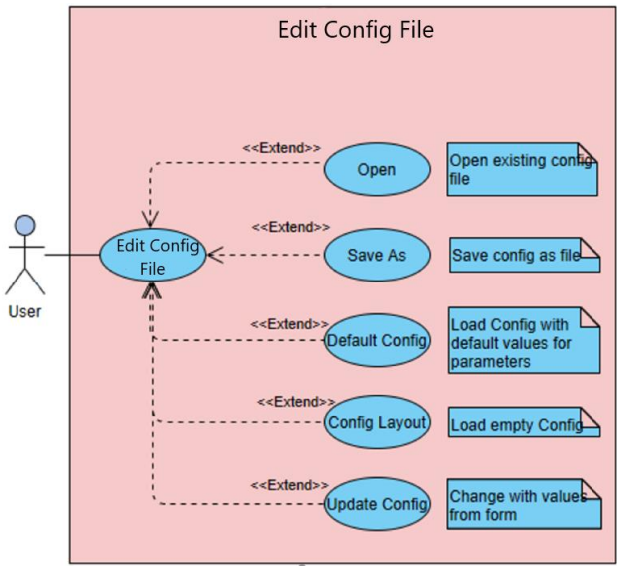


Figure 13 Config File Buttons Use case

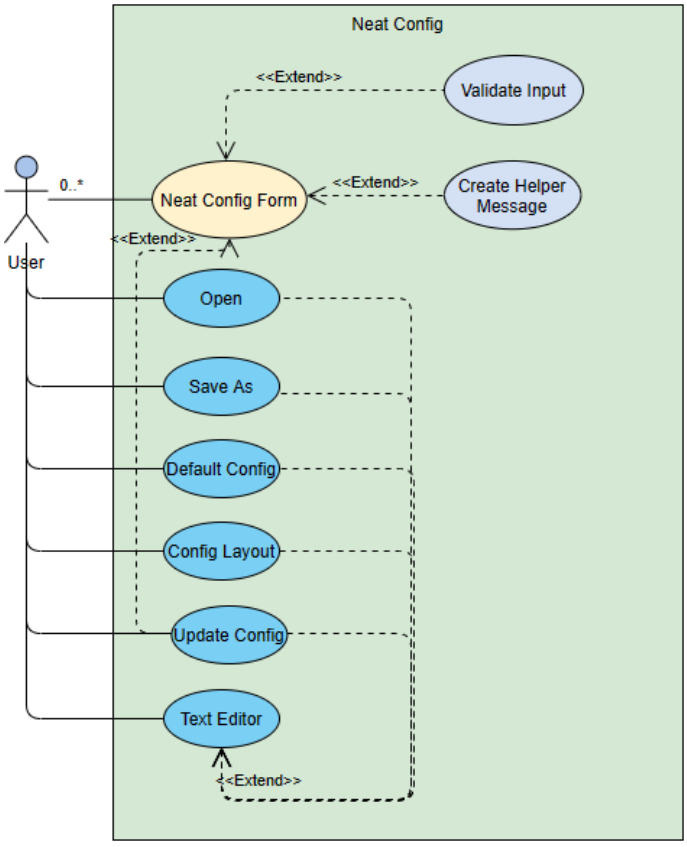


Figure 14 Neat Config Use Case

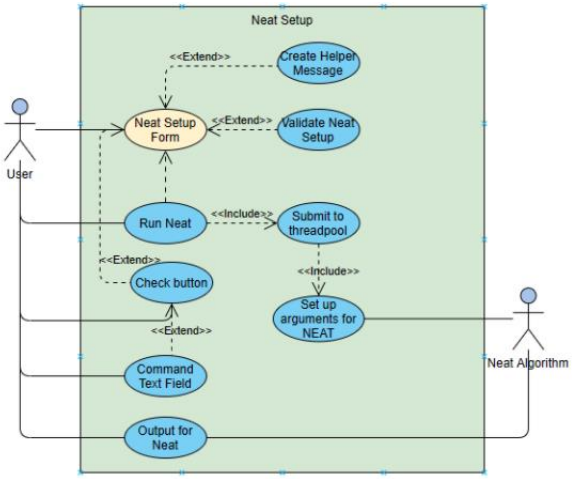


Figure 15 Neat Setup Use Case

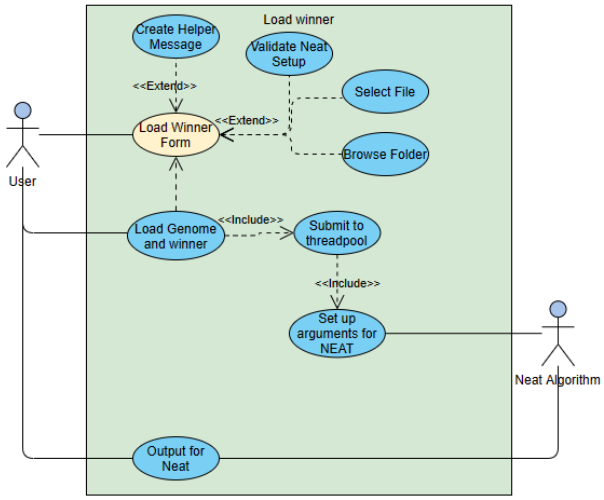


Figure 16 Load Winner Use Case

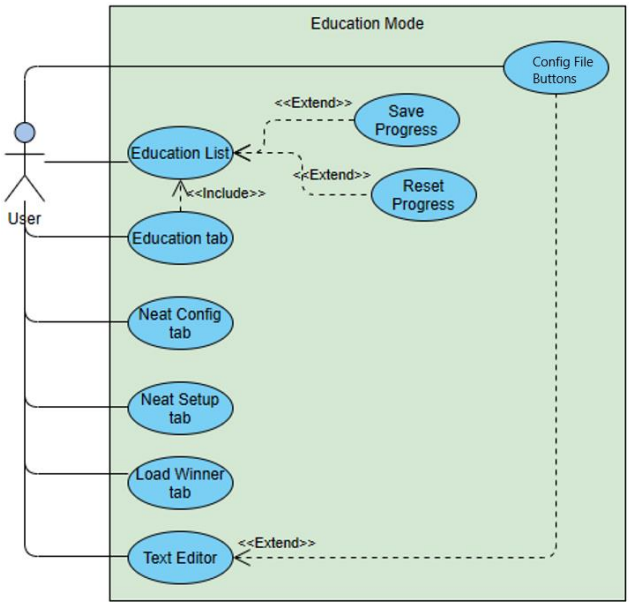


Figure 17 Education Mode Use Case

3.7.3 Activity Diagram

The activity diagrams are developed using the Lucid chart[8] web application (education plan).

3.7.3.1 Neat Config

The activities within Neat Config are a lot as there are many functionalities needed to make the program easy to use and include all functionalities. (See Fig. 18)

- Select Game – When a game is selected from the list, the game will be validated that it is indeed from the list, and it will try to return the inputs and output values for that specific game automatically onto the form (set the text of the text fields with the returned values) (User Story 10) from Editor)
- Input/Output Parameters – the user can manually write the input and output values for the game, or the user can get these parameters from selecting a game, and it will automatically generate the input and output based on the game
- Aggregation default and Activation Default – The algorithm will usually randomly set these parameters values if nothing is specified in the config file. For example, the user is given an option to choose a random generator to know in advance what values the algorithm is using, or if the random selector is not set to true, it will check if the value is valid.
- All Neat Parameters – all of the parameters have specific values they can be set to in the configuration file, so they all go through validation when you move the cursor out of the text box in order to provide the user with an idea of what values can be inserted and what values cannot be. Whenever a value has a default parameter, when moving our cursor out of the text box, if a wrong value is inserted or left empty, it will set the default value for that text field (User Story 8) from Editor)
    - Validation for the values is gathered from the Neat Python 0.92 Documentation (Welcome to NEAT-Python's Documentation! — NEAT-Python 0.92 Documentation, 2015) where each of the values has a specific input it accepts, so all values have exceptional validation cases

[8] Lucid Chart app - https://lucid.app/

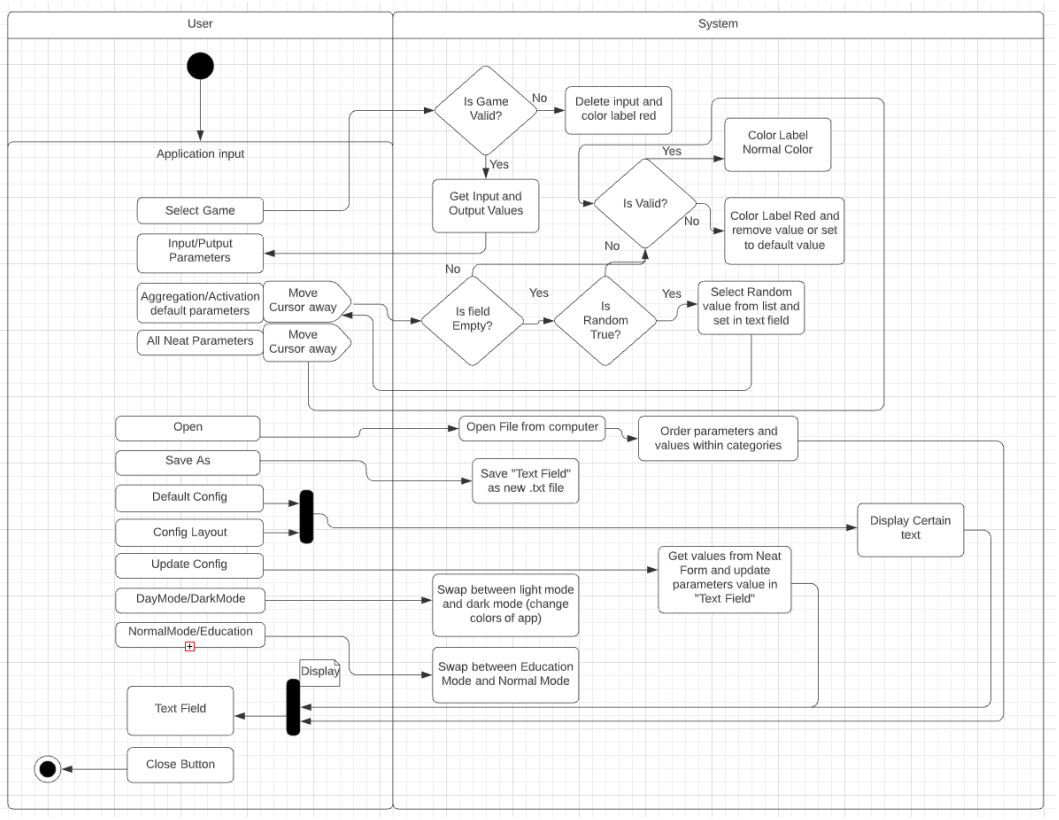


Figure 18 Neat Config Activity Diagram

3.7.3.2 Neat Setup

The activity diagram for Neat Setup shows a few triggered events (See Fig. 19):

- Game Selection Event – If the game chosen by the user is a 2DBox game, the extra text field “Runs per network:” will become available.
    - The Atari games do not have it as the algorithm crashes, and the 2DBox games have it because the algorithm would not run without it
- Receive Input – This event is triggered when the user presses a key on the keyboard and writes something in the winner file text field, so each character inputted into the text field is checked whether or not it is a unique character, and if yes, it is removed, so no issues appear when selecting the winner file. (User Story 6) from Neat Setup)
- On Choice Event – This event is triggered when the user selects “From Directory”, “Restore from checkpoint”, or “From Text Editor”.
    - The file selected is checked whether it contains no_fitness_termination = True
        - “From Directory” – Prompts the user to choose a file from their computer, which will get the full path directory
        - “From Text Editor” – Save the text in Text Field (Text Editor) as a .txt file and get the full path directory
    - No checks are performed on “Restore from Checkpoint”, which was restored from previous training and does not consider any other variable in the form selected at present but uses an older configuration when the checkpoint was saved.
- Checkbox event is triggered once the checkbox has been ticked and performs a check within the configuration file using the game selected by the user. The game selected is checked whether or not it exists in available games, and if it

does, it returns the input and the output for the game and displays it in the “Enter Command Text Box”.

- Run Neat Output – This event is triggered within the NEAT algorithm, so when the algorithm is run, the output produced by the Statistic Reporter is redirected to the Text Field found on the form so the user can see the progress.

The whole form is vital to starting the training of the NEAT Algorithm as it contains information that the user can modify to run NEAT. The start of the algorithm is launched from the button “Run NEAT”, where various checks are made before starting the algorithm.

The user stories 2) and 4) from Neat Setup can be seen in the Activity diagram. The 2) user story can be seen in the Neat Setup form, where various parameters can specify various parameters to modify how the algorithm will run. For example, as we can see, the output of Neat is outputted into the Text Field from the algorithm (output of the algorithm is redirected to the Text Field, where it usually would be seen in the terminal (User Story 4)).

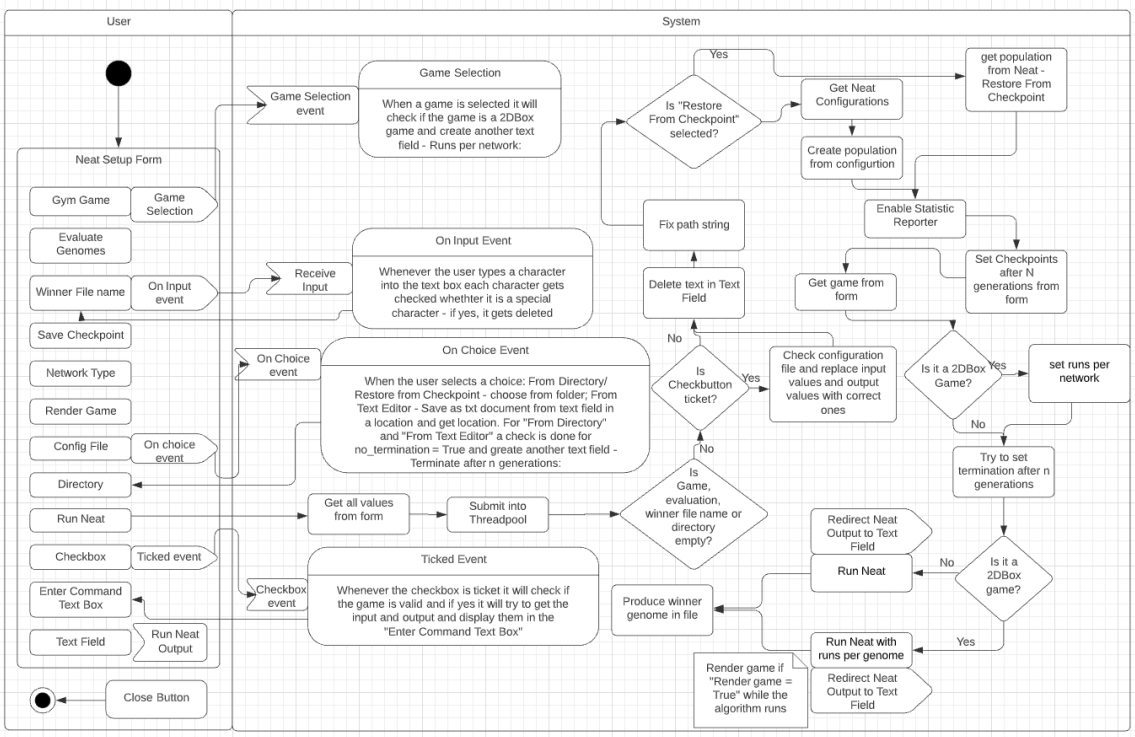


Figure 19 Neat Setup Activity Diagram

3.7.3.3 Load Winner

The activity diagram for Load Winner shows a few triggered events (See Fig. 20):

- Winner file name – triggered on input where when the user inputs a character, the software checks if it is a unique character and removes it.
- Select File – This will prompt the user to select the winner genome file from a folder to browse on their computer. (User Story 9) from Load Winner)
- Checkpoint(s) directory: Once the directory is selected from the prompt when Browse Folder Button has been clicked, the program will search for any checkpoints within that folder that match “neat-checkpoint”. Once at least one checkpoint has been found, the program will unhide another text field “Num of genomes in checkpoint:” (User Story 5) and 6) from Load Genome). This

enables the user to view the desired checkpoints within a for loop and outputs in the text field which genome is being run (User Story 7) from Load Genome)

- Config file – The config file will trigger an event to prompt the user to choose a file that the winner genome is saved.

Any of the parameters in the form are crucial for loading the winner genome and checkpoint(s) (optional), and after a study on what can be done with the Neat Algorithm, the parameters on the form have been implemented for a reason. (User Story 1) from Load Genome)

The “Show Winner Genome” (output the winner genome) and the output of the checkpoints being run and which genome within that checkpoint, depending on the number of checkpoints shown in the Text Field during each run. This satisfies User Story 8 from Load Genome, where the user can see what is loaded to understand better what is being shown while the game is rendered so the user can make comparisons between the generations.

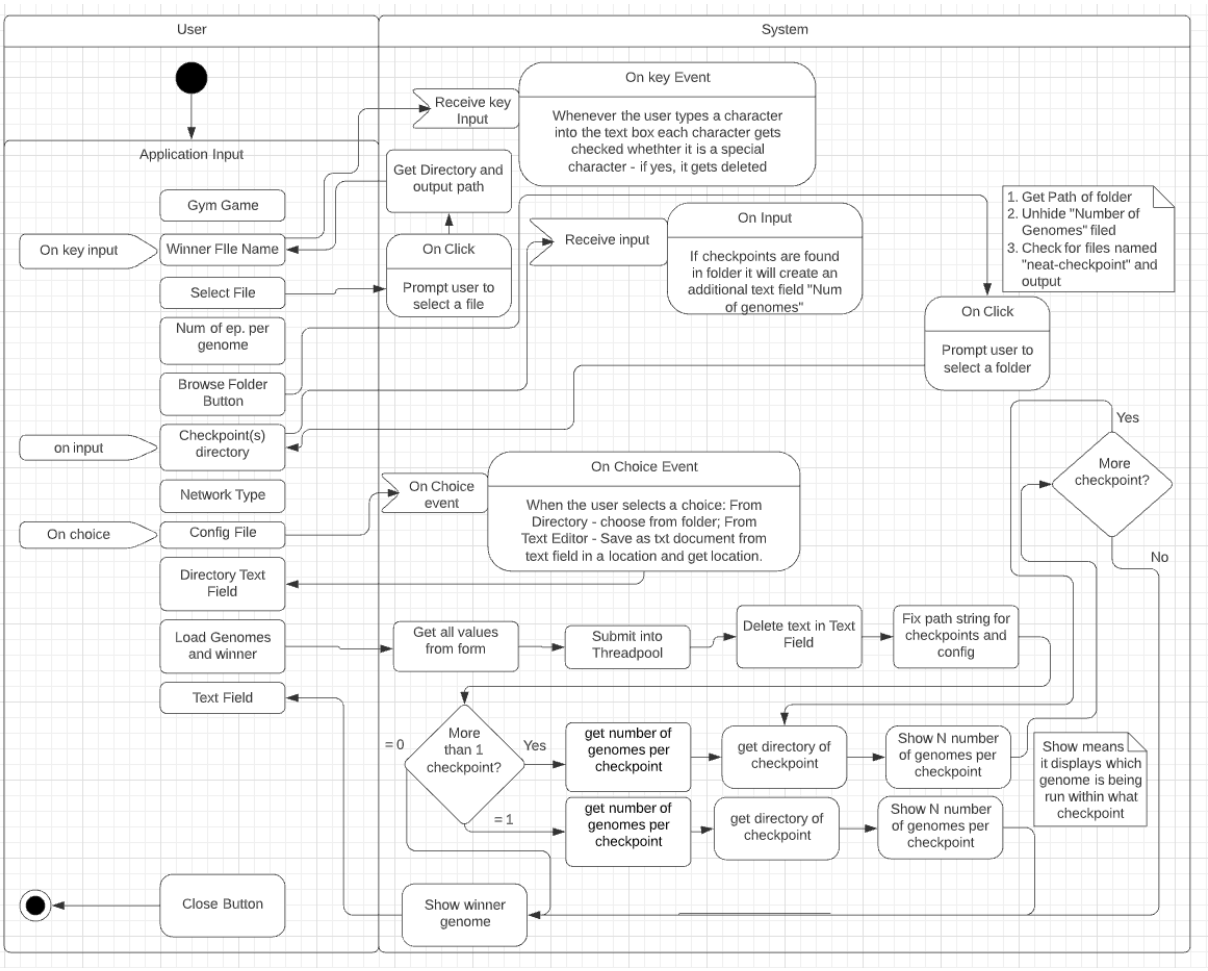


Figure 20 Load Winner Activity Diagram

3.7.3.4 Education Mode

The activity diagram for education mode shown in Fig. 21 shows the essential components of education mode where an event triggers a few functions:

- Choose Lesson – Choosing a lesson from the Education list triggers the program to load the appropriate lesson based on that choice and unlocks the next lesson. This event is designed to satisfy User Story 3 from Education mode, where the user has to go through the lessons one by one to proceed.

- Next Button in a lesson – In some of the lessons, a “Next” button will be generated so the user can see how to use the NEAT algorithm and set it up. After the user has completed these examples, the user should be able to configure essential features of the Neat Config file, start NEAT and load the winner genome alone. (User Story 4),5) and 9) from Education mode)
- On click (Save Progress button/Reset Progress button)
    - Save Progress: using this button, the user will be able to save which lessons have been open and save them in a log file which will be used next time the user loads the app. (User Story 6) from Education mode)
    - Reset Progress: using this button, the user will be able to reset the progress on the lessons having the lessons locked and requiring the user to view the lessons one by one. (User Story 7) from Education mode)

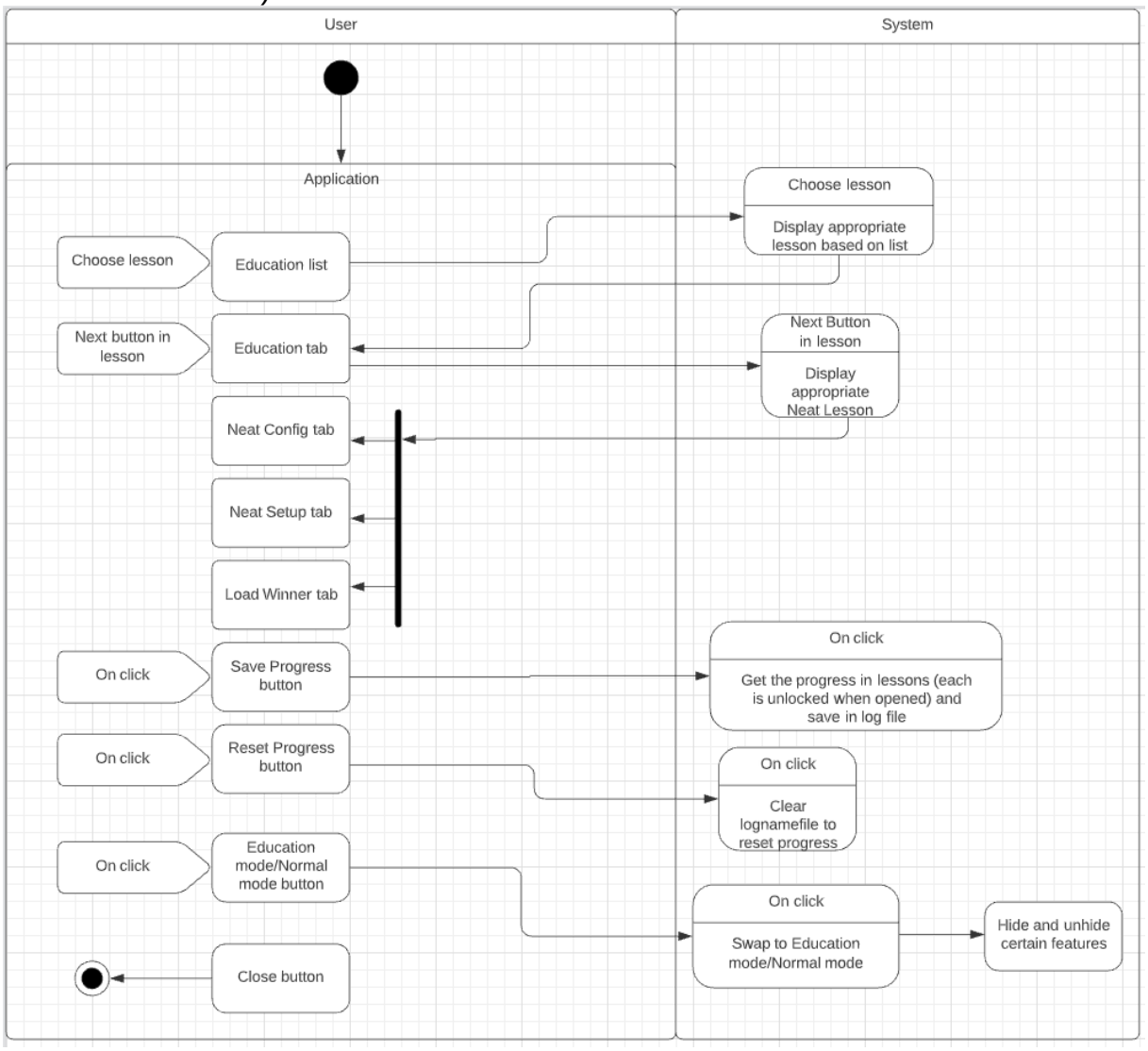


Figure 21 Education Mode Activity Diagram

3.8 Validating Requirements

Validating the requirements is an essential step as having invalid requirements may be time-consuming to rectify, but because the project is done following the agile methodology changing the requirements does not cost too much of an issue because changes can occur at any sprint, so that is why showing a working prototype of the

application during is sprint is essential – it shows the users what can the app do and if any changes need to be done after implementing the requirements set for the sprint.

4 Design and Implementation

4.1 Agile

The program's implementation was designed following the agile software development principles to build the program using quick cycles that allow instant production and continuous revision.
The development of the program was broken down into small pieces to achieve sprints that occur weekly. The tasks to do during each sprint were written in a Kanban log (Fig. 22), where some tasks have higher priority than others. Implementing the program in an agile way allowed many changes because feedback on the progress was given weekly, allowing time for any changes that needed to be made—executing several tasks each sprint allowed time to focus on the tasks for that particular sprint. The plan for the development of the program from the start can be viewed in Appendix 3 10.3, together with the new plan, which is how the program was implemented.

4.1.1 Kanban

The Kanban has been used to visualize the tasks that need to be done using the Trello application, and tasks that need to be done are in the backlog and then moved into the appropriate column to track progress (See Fig. 21). Access link to the board can be found in Appendix 3 10.2 Trello Board.

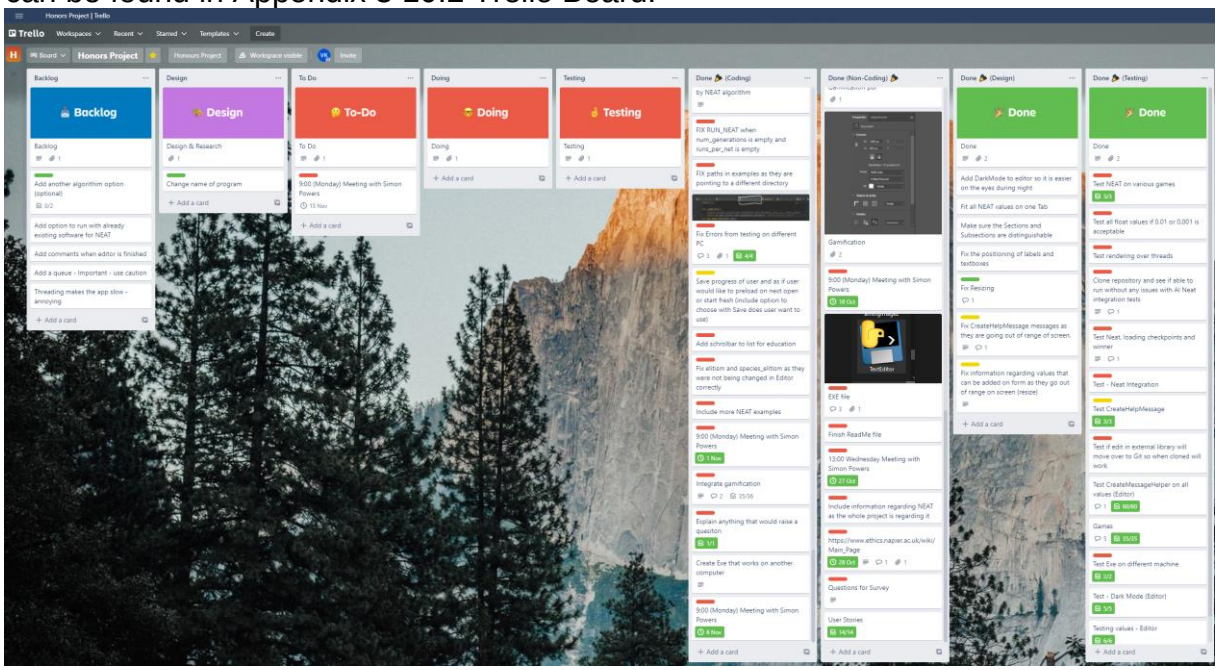


Figure 22 Kanban Board

4.1.2 Sprint

A particular set of tasks needs to be completed each spring to reach the sprint's desired goal. The sprint lasted one week where every week there is a sprint review where it is discussed what has been done and speak about any changes that might need to be done or the desired goal for the next cycle. During the sprint review, part of the sprint planning and sprint retrospective is done in order to answer the questions: “What work can get done in this sprint and how will the chosen work get

done?” and identify areas of improvement for the next sprint (REHKOPF, n.d.). The individual sprint meeting notes can be found in Appendix 3 10.1 Meeting Notes.

4.1.2.1 Sprint Planning

As mentioned above, part of the sprint planning is done during the sprint review. User stories are used to identify what needs to be done has been written in a clear, easy to understand the text. For the sprint to be successful, making each task as straightforward as possible is a must so there are no questions (RADIGAN, n.d.).

4.1.2.2 Daily Stand-up

During the daily stand-up, the following questions were answered to identify: what has been done, what needs to be done, and any issues: (RADIGAN, n.d.)

- What did I complete yesterday?
- What will I work on today?
- Am I blocked by anything?

4.1.2.3 Retrospective

During the retrospective, feedback on the current progress is given to understand what went well and what did not. If an issue is found, then a workaround needs to be constructed in order to fix it. Continued improvement has been gained from the retrospective as it helped understand the software's goals (RADIGAN, n.d.).

4.1.3 Version Control

Version Control is used for backup and restoring files that have been modified or deleted. Changes to the files can be tracked and retrieved to any previous version, making it a lot easier to test NEAT’s implementations and whether or not they work. Version control also enabled testing changes in an isolated area (branch) before merging into the main repository.
Version control is used from the start of the development of the application until the finish. Every task is pushed onto the branch as a separate commit so tasks can be tracked. A previous version can be restored if a significant issue appears within the application after a change has been made.
The distributed version control system is used to develop the application, where the whole history of all of the files is stored locally on the machine. Changes in the application are pushed to the server once confirmed that the change does not cause any issues.
No locking on the files is needed as only one developer is working on the whole project.

4.2 System Design

While no overall architecture was followed while developing the application, it mostly resembles the event-driven architecture style as the application processes data on events.

4.2.1 Function-oriented design

The application's main files (TextEditor.py), Run_winner.py and Neat_Single_Processing.py, are based on the function-oriented design. These files are the main structures of the application.

4.2.1.1 TextEditor.py

TextEditor.py contains the whole graphical user interface (Education Mode, Neat Setup, Neat Config, Load Winner, the main Buttons and the Text Editor itself).
The system was designed following the function-oriented methodologies due to the trivial graphical interface. Having the graphical user interface written in an object-oriented design should lead to easier maintenance or changes, as it groups specific data elements and functions into an object. However, a considerate amount of planning and reengineering must be considered as the application has many text boxes, each one having different functionality and different kinds of validation.

4.2.2 Object-Oriented Design

The Build_in_Console.py, CreateHelpMessage.py, CustonText.py, Education_Tab.py, Get_Directory_For_Neat.py, ProcessImage.py, StickyNote.py, Validate_Neat_Setup.py and ValidateInput.py are all constructed following the object-oriented design methodologies. Each object within the class has specific functions and data.

4.3 The application

The application integrates many functionalities that the NEAT algorithm has, including some automation functions that should make editing the config, training the neural network and loading the winner easier.

4.3.1 Editing a Configuration file

In order to train a neural network using the NEAT algorithm or load a winner file, a correct and compatible (See section 4.4.4.1 for issues with Configuration file compatibility) configuration file must be used.
NEAT Config tab provides the full functionality for a user to edit an existing or new configuration file. All the parameters that NEAT v0.92 uses can be found on the NEAT Config tab (See Fig. 23).

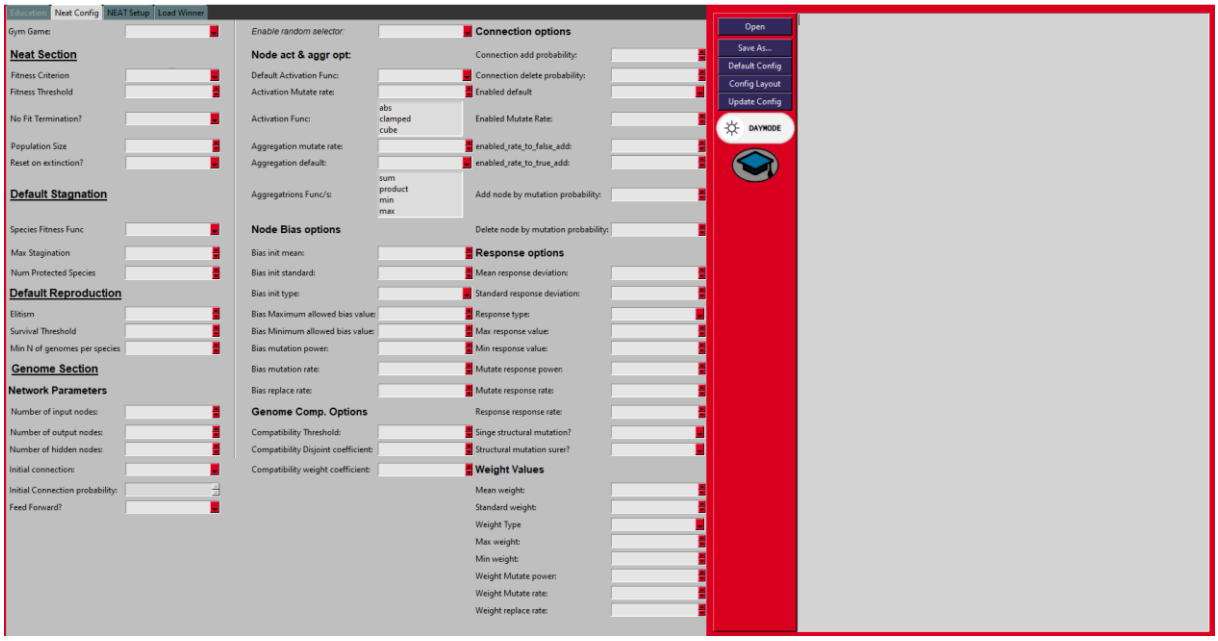


Figure 23 NEAT Config tab

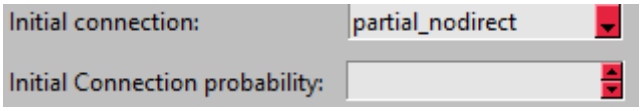


Figure 24 Enable probability text box

4.3.1.1 Loading and saving a configuration file

In order to give the user more freedom and satisfy User Story 1, 2, 3, 4 and 9 from Neat Config, the buttons have been implemented so they can always be used no matter where in the program a user is. (See Fig. 25)

Config Button	Purpose
Open	Open an existing config file (User Story 1 from Neat Config)
Save As	Save the text from Text Field Editor to a txt file (User Story 2 from NEAT Config)
Default Config	Load default values for parameters that have one (Took default values from official NEAT Guidance[9]) (User Story 3 from NEAT Config)
Config Layout	Generate a config file with needed parameters but with empty values (User Story 4 from NEAT Config)
Update Config	Transfer data from NEAT Config to Text Field Editor – needs to have Header Sections to work (User Story 9 from NEAT Config) (See Fig. 24)

[9] Official NEAT Guidance Config file Description - https://neat-python.readthedocs.io/en/latest/config_file.html

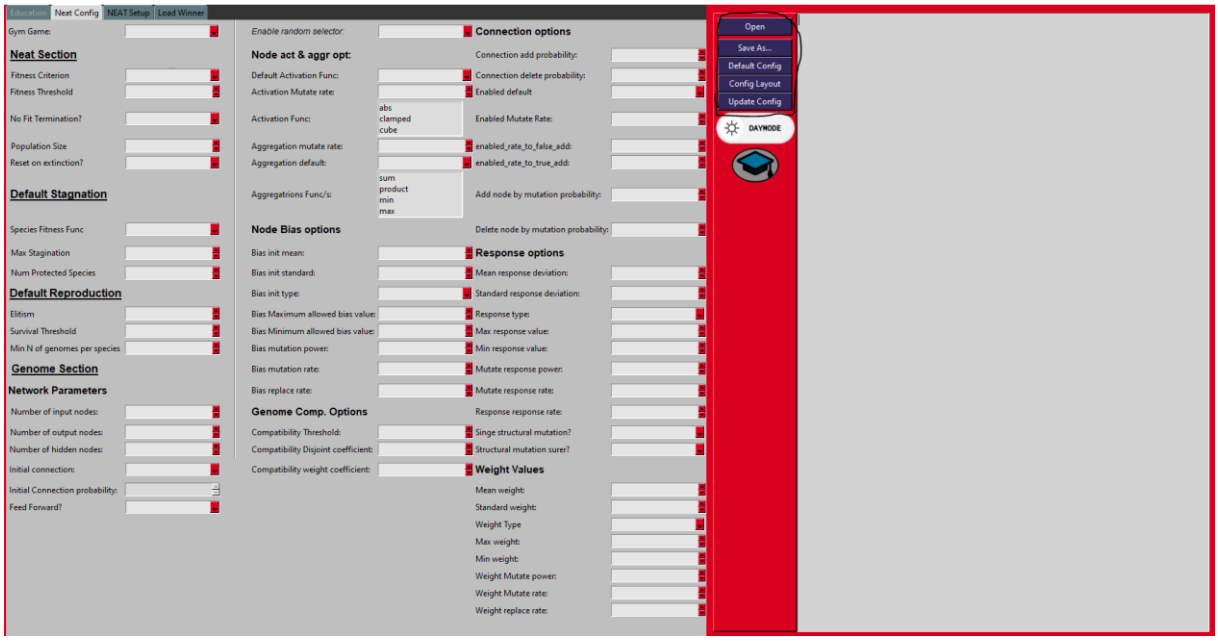


Figure 25 Config File Buttons

4.3.1.2 Opening an Existing Config File

When a user opens an existing config file it will structure it via additional sections within Default Genome for better readability. Additional sections separating the parameters:

- # Activation options
- # Aggregation options
- # Bias Options
- # Compatibility Options
- # Connection Options
- # Network Parameters
- # Response Options
- # Structure Options
- # Weight Options

Separating the sections improves the ease for the user to understand what the parameters are for, but it is used within the algorithm when it adds missing parameters, so it knows where to add them in the config file as all parameters are from a particular section.

4.3.1.3 Default Activation Function and Aggregation Function

Within the NEAT algorithm, if the default activation and a default aggregation function are not specified within the NEAT configuration file, NEAT will choose one at random, but the user will not later know which was chosen by the algorithm.
Because of that, the “Enable random selector” parameter was added, so when it is set to True, if the default activation function or default aggregation function is left empty, it will select one at random so the user can then update into the config file. (See Fig. 26)

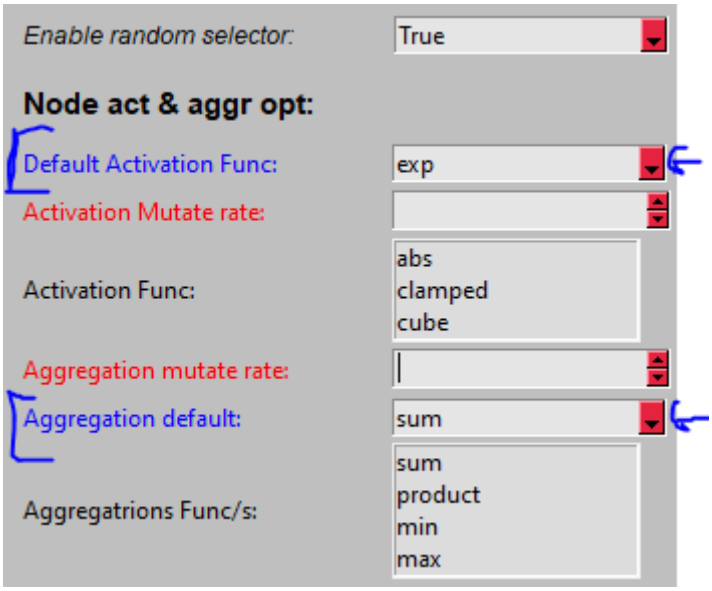


Figure 26 Randomly selected inputs (Random Selector = True)

4.3.1.4 Getting Input and Output values automatically

In order to make the application more robust and satisfy User Story 10, the feature of getting the input and output values automatically once a game is selected has been implemented. (See Fig. 27)
The input and output will not be generated if an invalid game is selected.

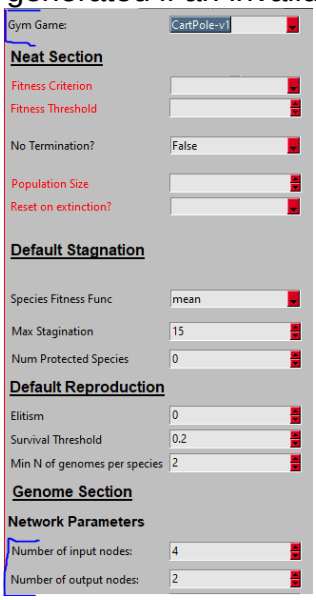


Figure 27 Automatic input/output fill depending on game selection

4.3.1.5 Leaving parameter empty

This is part of the validation process discussed in Test Outputs, but the table for NEAT config can be viewed in Appendix 4.

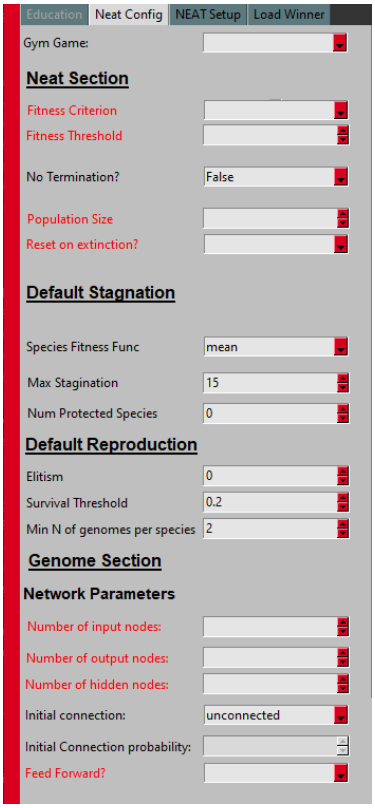


Figure 28 Input left empty - outcome

4.3.2 Training a neural network with the NEAT algorithm on a Gym game

The parameters (See Table 7) included for training a neural network are all the parameters that can be specified from a user/developer when choosing how to train the neural network. (User Story 2 from NEAT Setup)
The implementation considered what can be modified from a user to provide a more personal choice before the training is started. The graphical user interface (See Fig. 29) includes the form that a user needs to fill out to start the training.

Parameter and value	Meaning
Gym Game	Choose a game from a list of tested games (See Fig. 30)
Evaluate Genomes	This parameter specifies if the training will happen on one core (one genome at a time or more). Single-Processing is supported only in this program. NEAT provides the option for Multi-Processing, but it could not be integrated.
Winner File Name	Once the fitness_threshold is met, it will save the genome that has reached it so it can be viewed later
Save Checkpoint	This function gives the user the freedom to specify after how many generations a checkpoint should be generated. The checkpoint has two functionalities within the program: - View checkpoint to compare the progress of population for the specific generation - Resume training
Network Type	The user can choose between a Feed-Forward or Recurrent network. Value in config will be ignored. (See Fig. 33)

Render Game	Whether or not the user would like to see the game while it is being trained or not (User Story 3 from NEAT Setup)
Config File	This parameter is used so the user can choose from where to use the config file from: - File from a directory on the computer - From Text Editor (save and use path) - Resume from Checkpoint
Directory	This parameter is used so the user can see what the path that is chosen is
Check config file for input/output	This feature is an automation feature that, if enabled it will automatically change the input and output values of the config file based on the game chosen (User Story 5 from NEAT Setup)
Enter Command	This text field has two functionalities: - If “Check config file for input/output” is enabled, it will generate text that will show the user the changes that will be made in the config file (See Fig. 31) - Users can manually check what the input and output values for the game chosen via typing and pressing enter is: (See Fig. 32) o print(env.action_space) --- Output o print(env.observation_space) --- Input
Runs Per Network	This parameter is enabled only if the game is chosen in a Classic Control or Box2D game (See 4.3.3.1 section). The parameter specifies how many episodes will a genome be trained on before producing the fitness for the genome. (See Fig. 30)
Text Field “See the Evolution of genomes while running NEAT.”	This text field is used so the user can see the output of NEAT while it is running. (User Story 4 from NEAT Setup) (See Fig. 34)
Run NEAT	- Start the algorithm (the training) (See Fig. 36) - If validation is passed, all the parameters needed to load the winner are saved in logfilename.log (User Story 3 from Load Winner) (See Fig. 35)
Terminate after num of generations	After the config file is selected, the program scans it for no_fitness_termination = True, and if it is found, it will display this option. Choosing a number will determine for how many generations will the neural network train. (See Fig. 37)

Table 7 NEAT Training a Neural Network Parameters

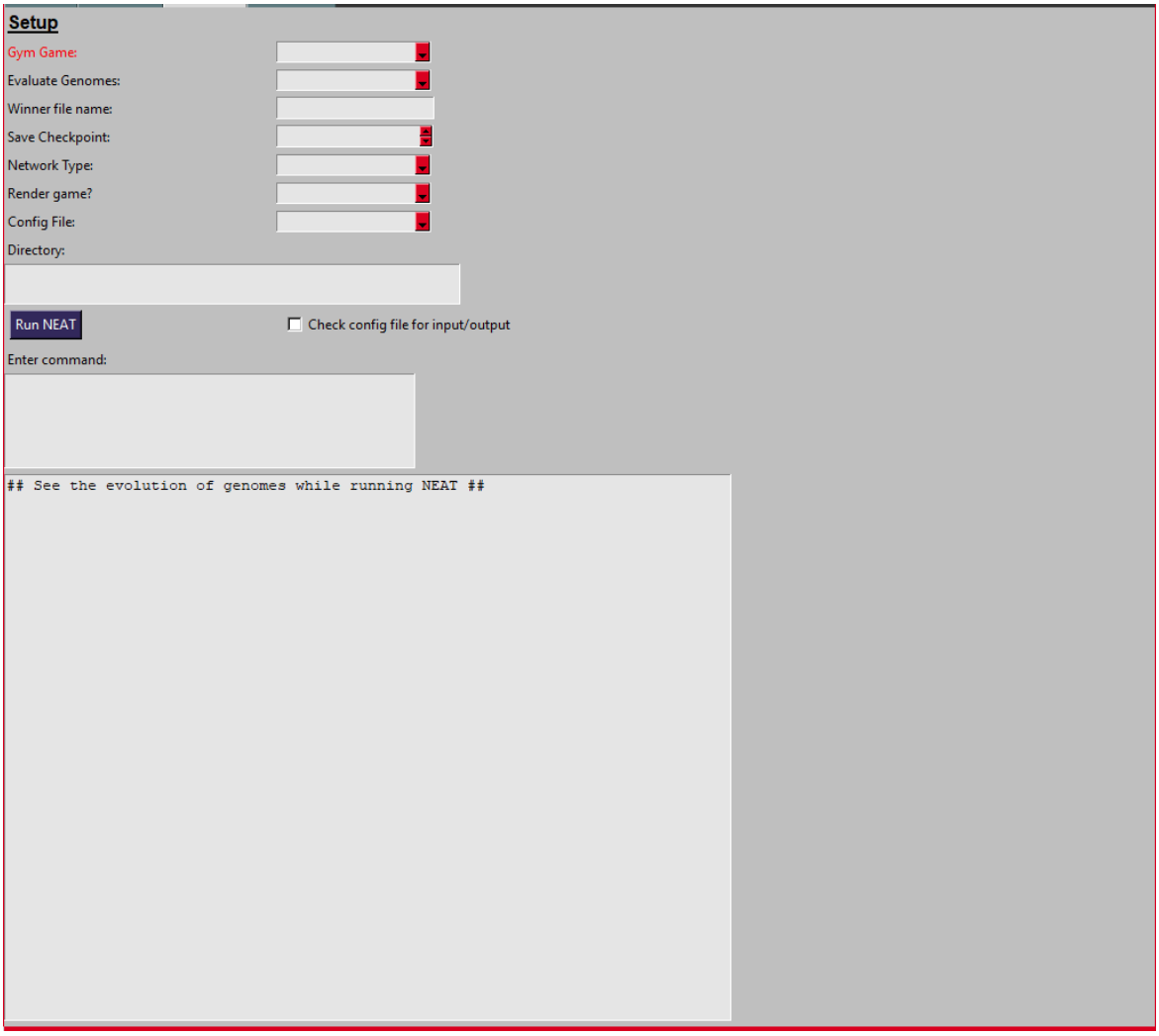


Figure 29 NEAT Setup (Train a neural network)

Figure 30 Runs per network reaction on Gym Game selection

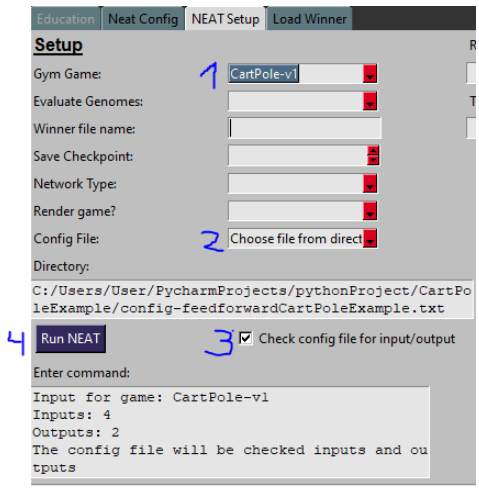


Figure 31 Enter Command Text Field reacting to the checkbox

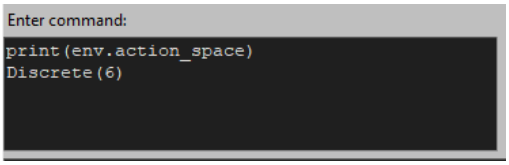


Figure 32 Manual input in Enter Command text field

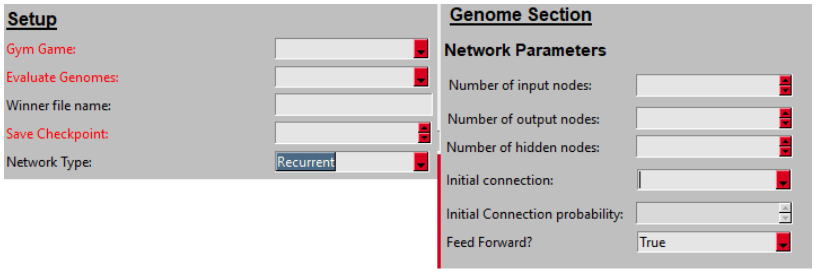


Figure 33 Network type Setup vs Config

```
****** Running generation 0 ******

Population's average fitness: 18.40000 stdev: 24.47039
Best fitness: 150.00000 - size: (2, 8) - species 1 - id 64
Average adjusted fitness: 0.073
Mean genetic distance 1.287, standard deviation 0.294
Population of 100 members in 1 species:
   ID   age  size  fitness  adj fit  stag
  ====  ===  ====  =======  =======  ====
     1    0   100    150.0    0.073     0
Total extinctions: 0
Generation time: 0.508 sec
Saving checkpoint to neat-checkpoint-0
Saving checkpoint to neat-checkpoint-0

****** Running generation 1 ******

Population's average fitness: 34.80000 stdev: 57.72088
Best fitness: 500.00000 - size: (2, 8) - species 1 - id 178
Average adjusted fitness: 0.054
Mean genetic distance 1.493, standard deviation 0.325
Population of 100 members in 1 species:
   ID   age  size  fitness  adj fit  stag
  ====  ===  ====  =======  =======  ====
     1    1   100    500.0    0.054     0
Total extinctions: 0
Generation time: 1.250 sec (0.879 average)
```

Figure 34 See the evolution of genomes while running NEAT text field (training in progress)

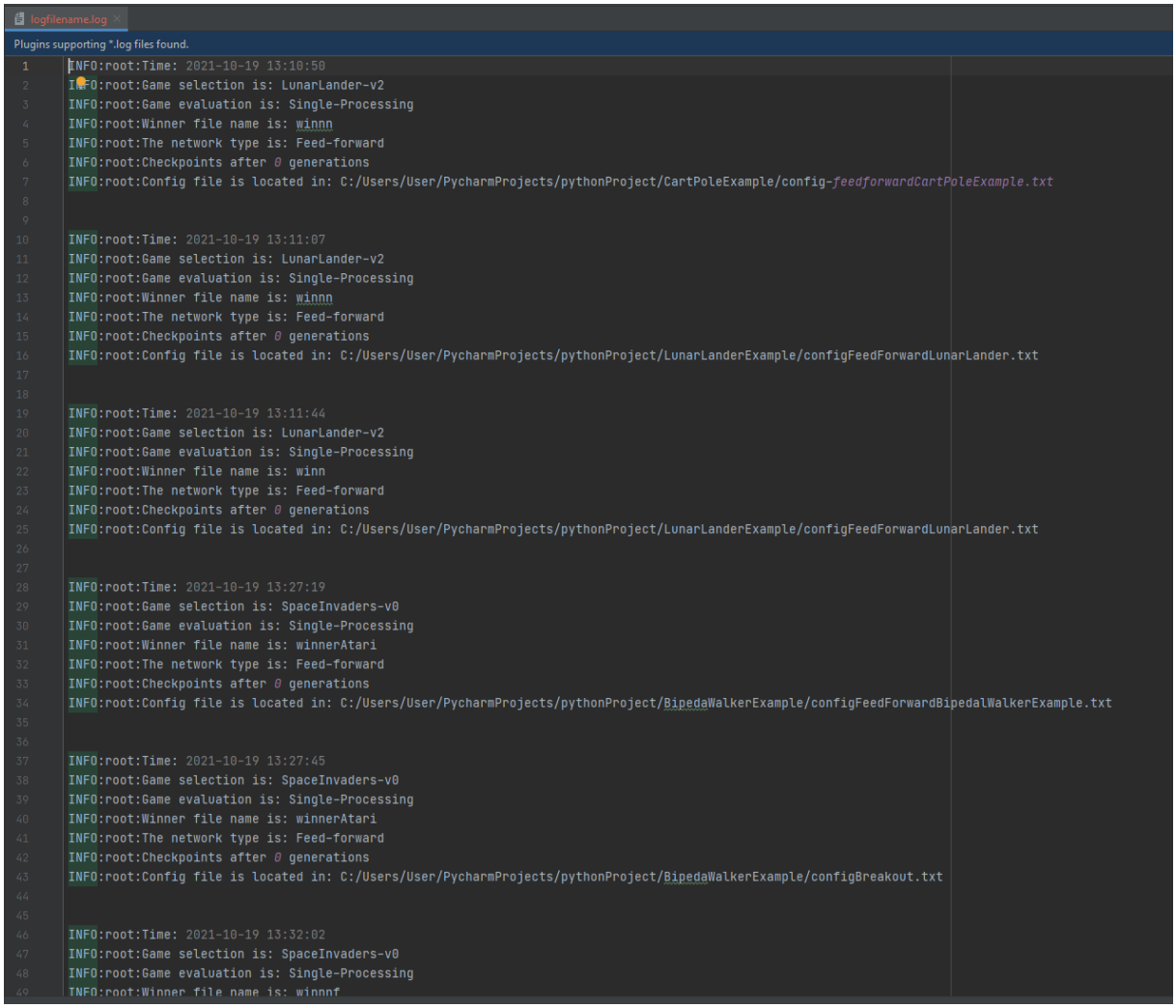


Figure 35 Example of logfilename.log

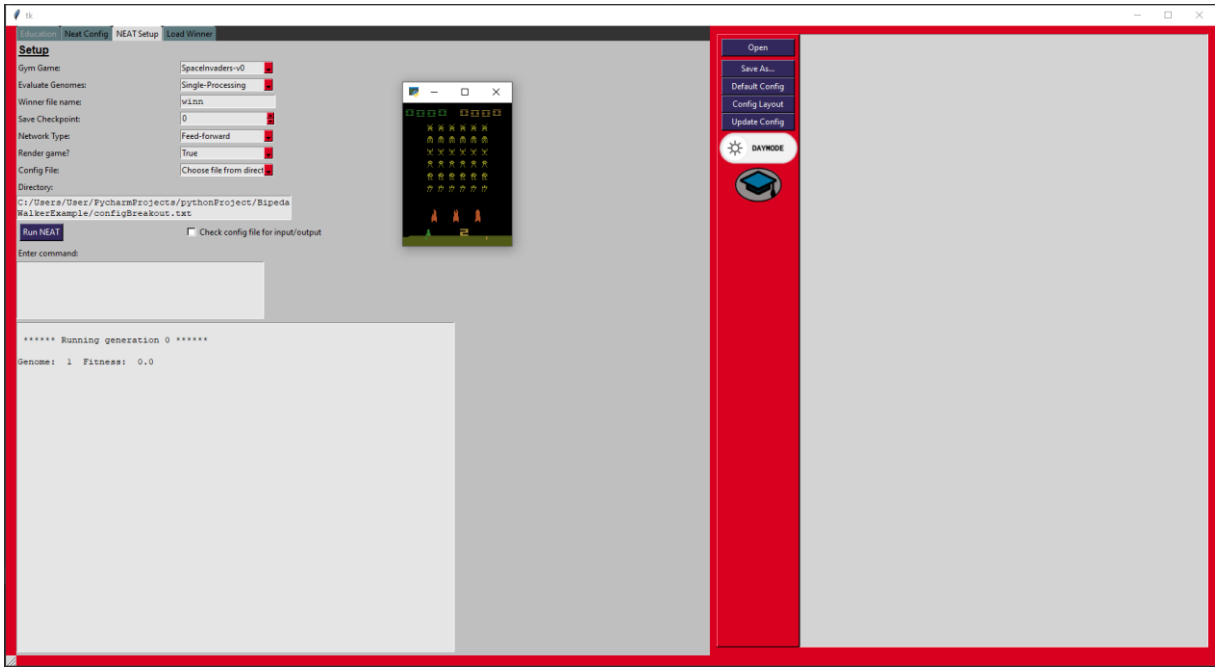


Figure 36 Training neural network to play Space Invaders

4.3.2.1 No Fitness Termination

If in the configuration file no_termination = True, it will display another form input that the user can specify after how many generations to stop the training. This is a clever feature as the network might train for a long time and either find a winner genome (reached threshold) or reach complete extinction. (See Fig. 37)

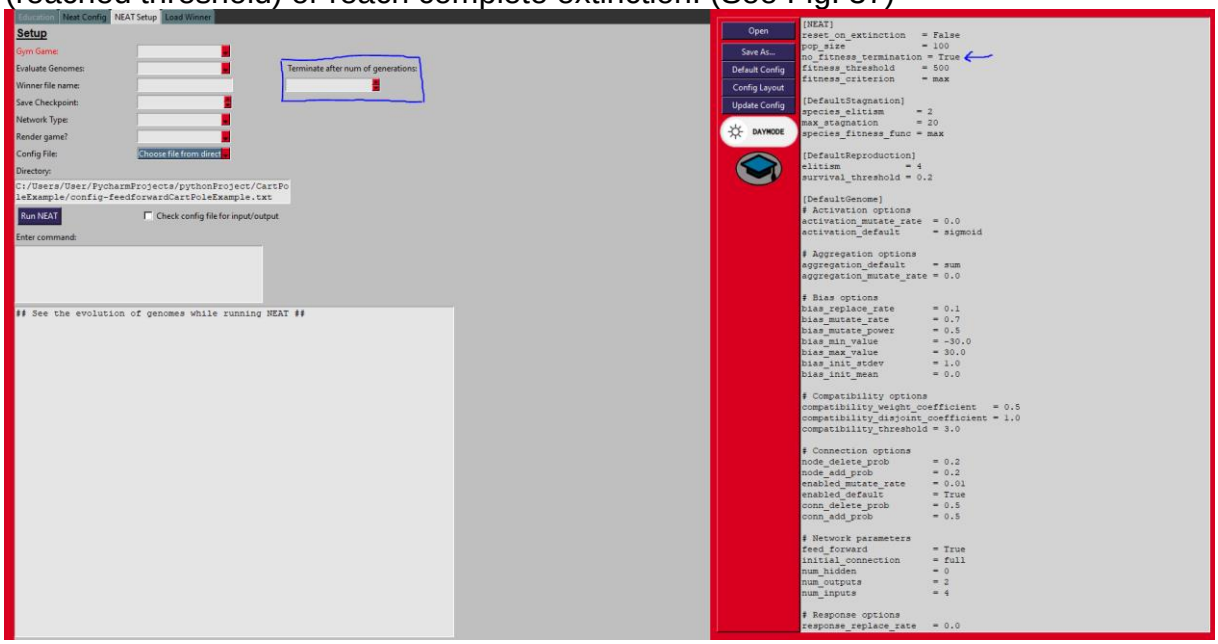


Figure 37 No Fitness Termination

4.3.3 Loading the winner genome and checkpoints

The sole purpose of training a neural network is to see the outcome of the training then. That is why the Load Winner tab is developed.
The implementation of this tab considered all parameters that can be modified and specified before loading the winner. (See Table 8 and Fig. 38)
The NEAT algorithm has been modified to allow a user to view the generation's progress saved as a checkpoint (See 4.3.4.2 for implementation of NEAT changes). An example of the loading of checkpoints can be seen in Fig. 40.

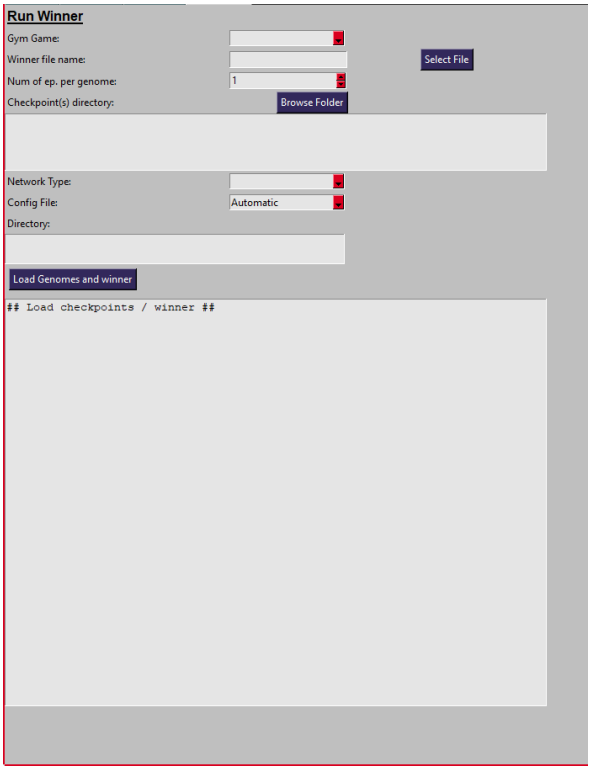


Figure 38 Loading the winner genome and checkpoints

Parameter and value	Meaning
Gym Game	The game used for the training
Winner file name	The winner genome that the user would like to see
Select File	Choose winner file from a directory (User Story 9 from Load Winner)
Num of ep. Per genome	Test the winner genome of N number of episodes (User Story 4 from Load Winner)
Checkpoints(s) directory (Browse Folder)	The button prompts the user to choose a directory where it should search for the checkpoints. Checkpoints are saved as neat-checkpoint-n, so the algorithm searches in a directory for neat-checkpoint
Checkpoints(s) directory (Text Field)	The checkpoints found in the specified directory from “Browse Folder” are displayed in the text field, and the user can delete any of the checkpoints that are not wanted in the viewing process
Network Type	The type of training performed on the winner genome
Config File	This parameter is used so the user can choose from where to use the config file from: - File from a directory on a computer

	- From Text Editor (save and use path) - Resume from Checkpoint
Directory	This text field is used so the user can see what the path that is chosen is
Text Field “Load checkpoints/winner.”	This text field is used so the user can see what is being executed on the screen. (User Story 8 from Load Winner)
Load Genomes and winner	This button, once clicked, starts showing the checkpoints (if any specified) and the winner genome N times (See Fig. 40)
Num of genomes in checkpoint	Enabled if “Checkpoint(s) directory (Text Field) is not empty. How many genomes to display from generation as if the population is 100 it will display 100 episodes (See Fig. 39)

Table 8 NEAT Load Winner/Checkpoints Parameters

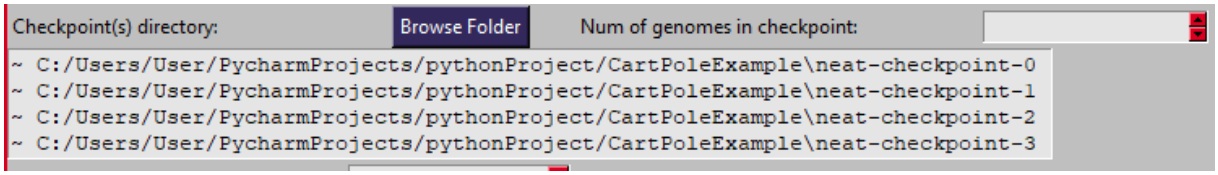


Figure 39 Number of genomes to show in a checkpoint

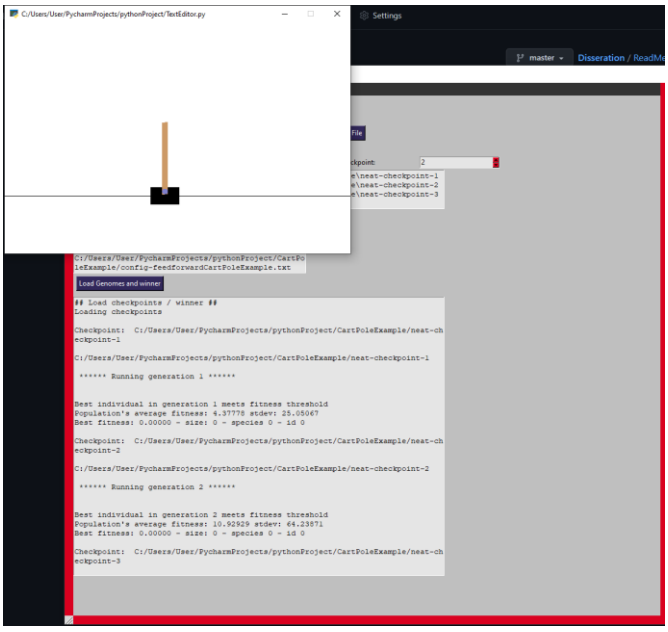


Figure 40 Viewing Checkpoints and then the winner

4.3.4 The transition from NEAT Setup to Load winner tab

When a user trains a neural network and has found the winner genome that meets the fitness_threshold, the user wants to see the outcome rendered, meaning what the genome looks like when trained.
If the NEAT Setup tab is filled out and the user clicks on the Load Winner tab, the values from NEAT Setup will transfer over automatically to Load Winner to save the user the hustle to write everything up again. This feature was included following User Story 2 from Load Winner to make the application more automatic. (See Fig. 30)

4.3.5 Helper Messages

Helper messages are implemented to satisfy User Story 6 from NEAT Config but have also been included throughout the program to help the user better understand what parameter is used for what.
The helper message appears below the label or text box when the mouse cursor is over one of them. (See Fig. 41)

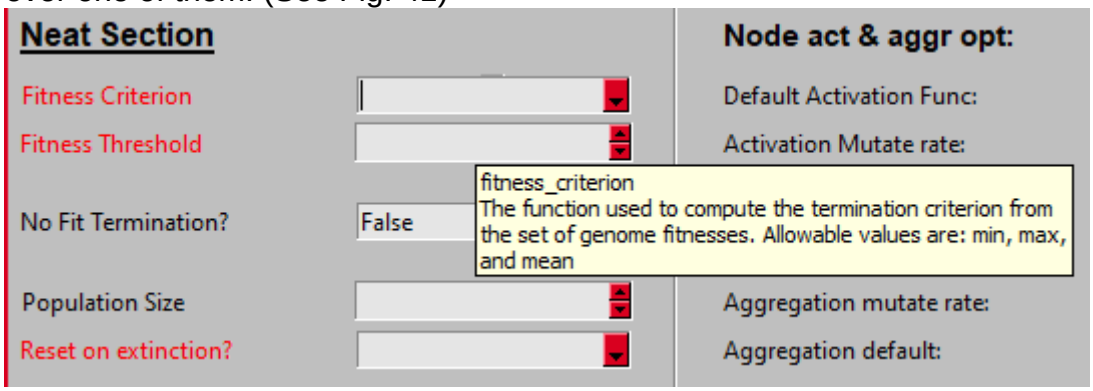


Figure 41 Helper Message over fitness_criterion

4.3.6 Games used in application

The games that can be used for training a neural network are all integrated from Open AI Gym[10]. The Gym library is a set of test problems – environments – that can develop a reinforcement learning algorithm. (OpenAI, n.d.)
To get the library Gym working, a compatible version of Python needed to be used. The official documentation was initially followed to install the library Gym and test if it works. Issues relating to the games (non-Atari) were found with the latest version, so Gym 0.18.0 is used throughout this project.

4.3.6.1 Classic Control and Box2D

In order to implement NEAT with the Gym Classic Control and Box2D environments, various examples that use NEAT or another reinforcement learning algorithm (Brockman, 2015; Kinsley, 2017; Omelianenko, 2019, Chapter PyTorch NEAT usage example; Open Source, 2018) were taken into account when developing the algorithm. It is purely left to NEAT to decide what action the network should take to reach the fitness threshold.
A working example can be seen in Fig. 42 for CartPole-v1. In order to get Classic Control games and Box2D games to work, runs (episodes) per network needed to be added into the implementation to get the algorithm executing; otherwise, an error

[10] Open AI Gym - https://gym.openai.com/

message would appear without it. Runs Per network means that a single genome will be trained five times in this example to produce its fitness and then move on to the next genome.

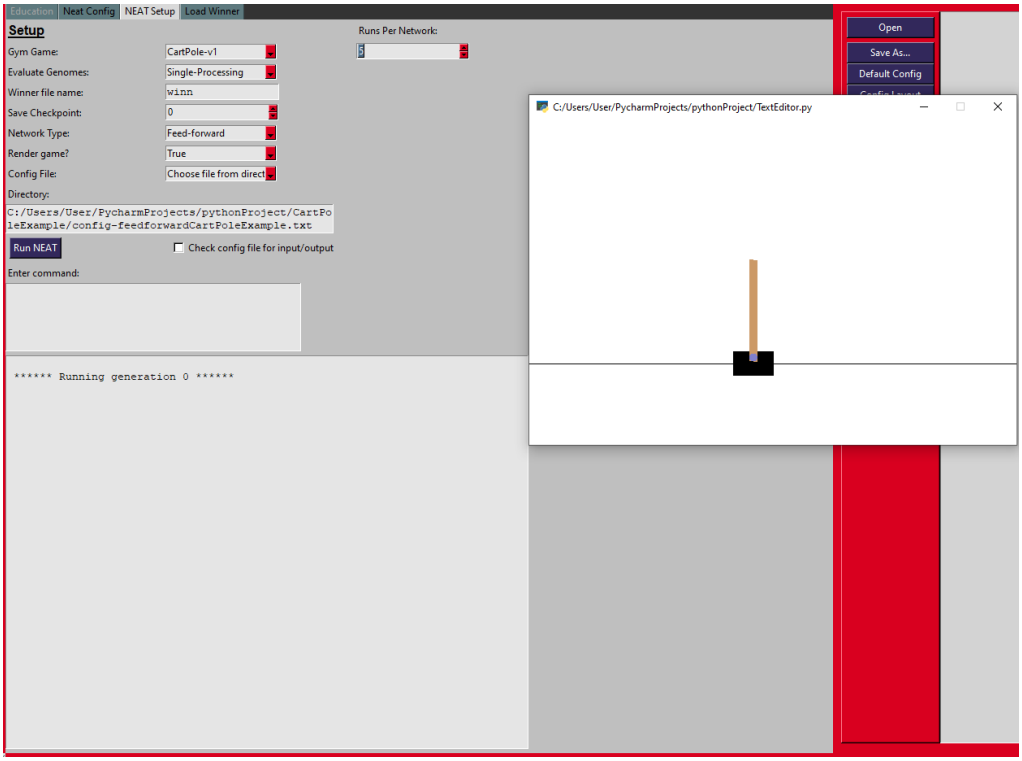


Figure 42 NEAT Neural Network Training Classic Control

4.3.6.2 Atari Games

Importing and using the Atari games was a challenge due to the confusing compatibility versions and pre-requirements. Section 4.3.3.2 discusses the issues in more detail.

Implementing the Atari-py games was more challenging than the classic control and Box2D games as the observation is an RGB image of the screen where the array is quite significant. Image processing is performed to make the training more straightforward and faster for the neural network:

- the length and width are divided by 3
- the image is converted into greyscale, so there are few colours for the algorithm to figure out.
- The observation is then put into a one-dimensional array that the neural network can understand, and the rest is left as usual.

For the Atari games, the “Runs per network” was not implemented as the algorithm broke after the first generation, and it is not necessary to have it to train a neural network successfully. (See Fig. 36)

4.3.7 Education Mode

The program's objective is for the user to be able to use the NEAT algorithm to train a neural network, see the winner genome, and compare the progress in the checkpoints. During the learning progress, the user is provided with a chatbot to see a different learning algorithm, questions that the user must answer and a guided walkthrough. Huang & Soman, 2013)

4.3.7.1 Graphical User Interface of Education mode

The graphical user interface of Education mode can be seen in Fig. 43, where it can be seen that Education mode is an extension of the default application. Education has been developed using features from gamification (Wahid & Wahid, 2018; Huang & Soman., 2013). The extension consists of another frame that consists of a list of lessons and “Save Progress” and “Reset Progress” buttons.

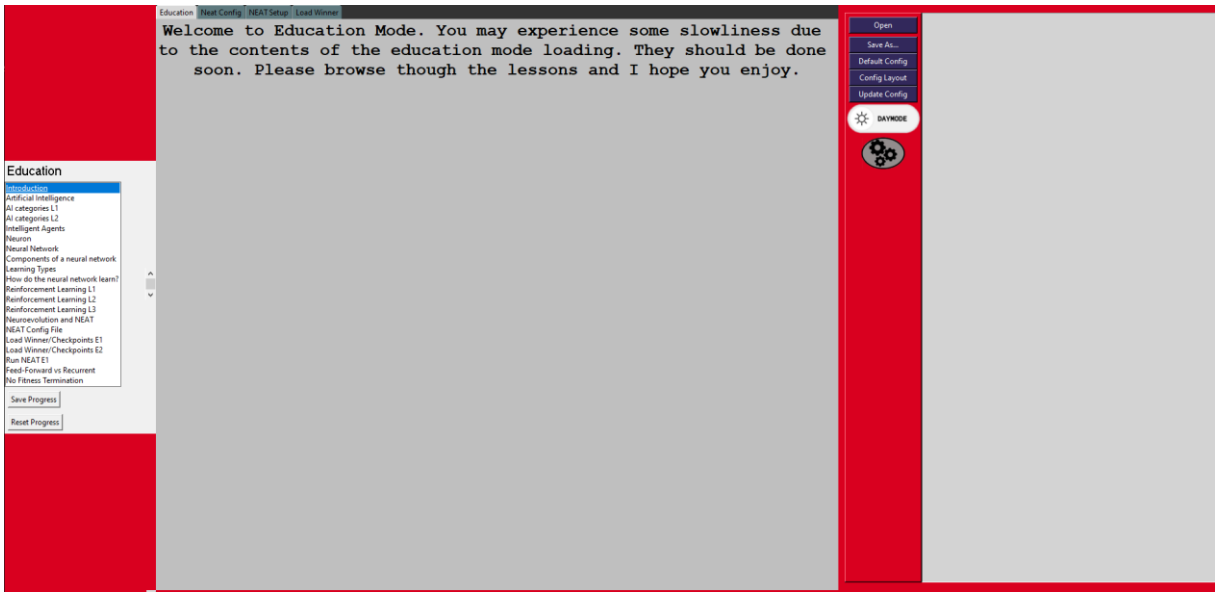


Figure 43 Education Mode GUI

4.3.7.2 Unlocking Lessons feature

A lesson is locked until you view the appropriate lesson for the Education Mode to be a guided learning path. Trying to view a further lesson before viewing the previous one would not work, as a message will show that a specific lesson needs to be referred to before viewing the one chosen. (See Fig. 44)

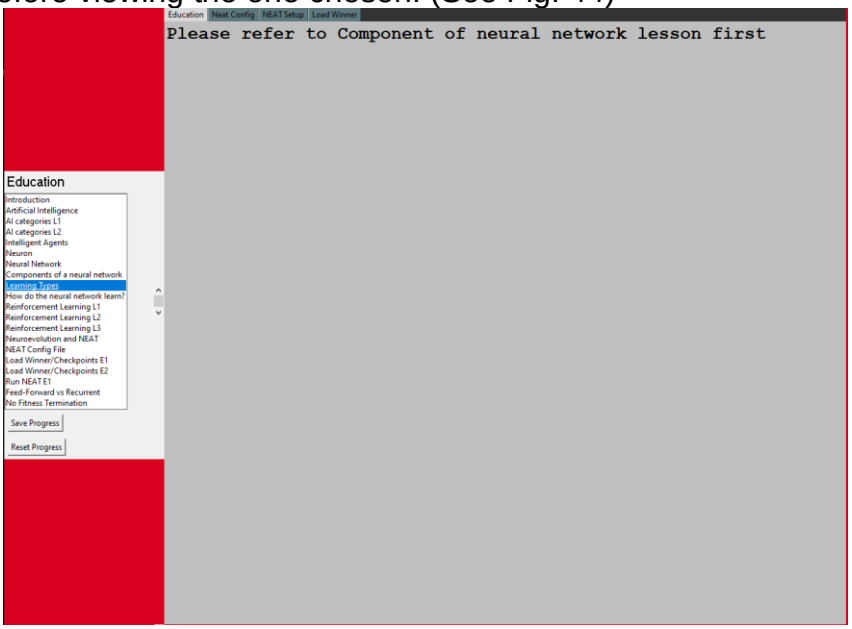


Figure 44 Locked Lesson Example

4.3.7.3 Save Progress / Reset Progress

The “Save Progress” button is implemented when a user clicks it saves which lessons have already been viewed, so when the application is relaunched, you can continue with the progress and do not have to go through everything again. The progress is saved in a progresslogfile.log. When an attempt to open a lesson, the lessons that already have been viewed are unlocked. (See Fig. 45)

```
Introduction_tab: False
Artificial_intelligence_tab: True
AI_Categories_tab: True
AI_Categories_tab_2: False
Intelligent_Agents_tab: False
Neuron_tab: False
Perceptron_tab: False
Components_of_NN: False
Learning_Types: False
Neat_config_choice: False
LoadWinner: False
LoadWinner2: False
Run_Neat_choice: False
How_Lean_NN: False
ReinforcementL1: False
ReinforcementL2: False
ReinforcementL3: False
Examples: False
Feedforward_vs_Recurrent_tab: False
No_Fitness_termination_tab: False
Atari_Example_tab: False
```

Figure 45 progress_logfile.log

The “Reset Progress” button will reset the progress, so you will have to view the lessons sequentially to unlock them when launched again.

4.3.7.4 Sticky Notes

The sticky notes are constructed via an object-oriented class that is easy to integrate into the application. The purpose of the Sticky Notes is to provide further information, so the format of the application does not change as labels, and text boxes have to be created and then hidden or removed. Furthermore, as discussed in section 4.4.2.2, displaying significant bits (text bigger than one line) can be challenging because the label widget is constructed.
For each of the walkthrough lessons, a sticky note is generated to further the user's knowledge. (See Fig. 46)

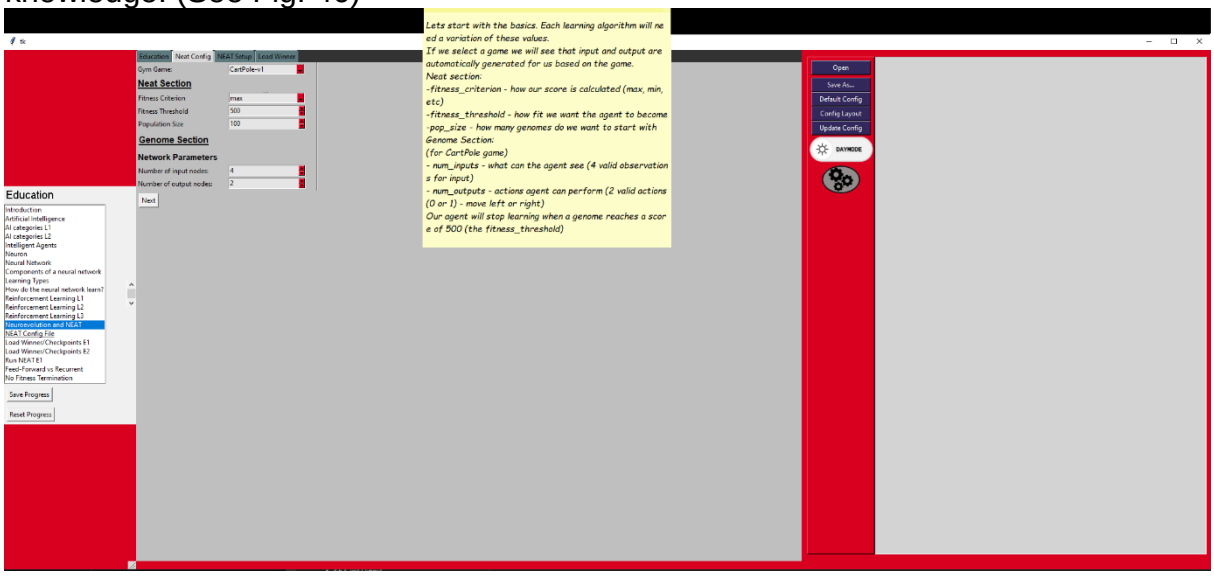


Figure 46 Sticky Note example

4.3.7.5 Lessons to read and (on some) answer questions

The lessons within the application include basic knowledge for Artificial Intelligence, the different learning types, Chatbot (an example of Supervised Learning), Reinforcement Learning, NEAT, Feed-Forward vs Recurrent and NEAT examples. Lessons have been constructed as images using a text editor like Word or Evernote and then created as imaged lessons using Photoshop and further edited using Photo3D as Tkinter does not provide good functionality with label display (See 4.4.2.2). See Table 9 for the lessons included in the Education mode that should further the user's knowledge in Artificial Intelligence. The table is missing the lessons that include going through the application explaining how to use it as they would be covered separately. Some of the lessons have questions that the user can answer. The lessons are inserted into Appendix 4 in order to view them without having to launch the application.

Lesson	Reference (the lesson is drawn on the material from)
Artificial intelligence	(Russell & Norvig, 1995, Chapter 1 Introduction; Taulli, 2019, Chapter 1 AI Foundations; Paul Mueller & Massaron, 2018)
AI Categories L1	
AI Categories L2	
Intelligent Agents	(Russell & Norvig, 1995, Chapter 2 Intelligent Agent)
Neuron	(Russell & Norvig, 1995, Chapter 19.1 How the brain works)
Neural Network	None
Components of a neural network	Figure Source: (Adapted from Russell & Norvig, 1995, Fig 19.4; Daume III, 2012, Chapter the Perceptron Fig. 3.2);
Learning Types	Sources for implementing table: (Adapted from Russell & Norvig, 1995, Chapter 18; Russell & Norvig, 1995, Chapter 20; Brown, 2014, pt. Neural Networks for Prediction)
How do neural networks learn?	(Hurbans, 2020, Chapter 9 Artificial Neural Networks)
Reinforcement Learning L1	(Hubans, 2020, Chapter 10 Reinforcement Learning With Q-Learning; Morales, 2020, Chapter 2 Mathematical Foundations of reinforcement learning)
Reinforcement Learning L2	
Reinforcement Learning L3	
Neuroevolution and Neat	(Stanley & Miikkulainen, 2002b; Xin Yao, 1999; Hubans, 2020, Chapter 4. Evolutionary Algorithms; Mirjalili 2018, p. 54; Örkcü & Bal, 2011)
Feed-Forward vs Recurrent	Sources for implementing table: (Adapted from What Are Recurrent Neural Networks?, 2021; Russell & Norvig, 1995)
NEAT Running Generation	(Omelianenko, 2019, Chapter PyTorch NEAT usage example)
NEAT Results from generation	
NEAT Found Winner	

Table 9 Lessons included

4.3.7.6 Lessons that walk through the application (with To-do Tasks)

The lessons are shown in Table 10 that walk the user through the application and explain the essential features of NEAT while showing how to use the application to edit a config file, train a neural network and see the outcome.
The lessons are all guided with a “Next” (See Fig. 47) button that automatically redirects the user to the appropriate tab once clicked. On the specific tab, the user will get a Sticky Note with some information regarding what is viewed and To-Do tasks to get the algorithm running for the specified purpose.

Lesson	Purpose
NEAT Config File	Explain to the user essential features of the NEAT config
Load Winner/Checkpoints E1	See an already trained neural network with the values from the previous example
Load Winner/Checkpoints E2	See how to load checkpoints and winner
Run NEAT E1	Explain and show to the user how to train a neural network using NEAT
No Fitness Termination	Explain and show the usage of and how to use no_fitness_termination
Atari Example Recurrent	See an Atari example on how to get inputs and outputs and then load trained recurrent neural network
Atari Example Feed-Forward	See the same example with a Feed-Forward Neural Network

Table 10 Lessons (Walkthrough app)

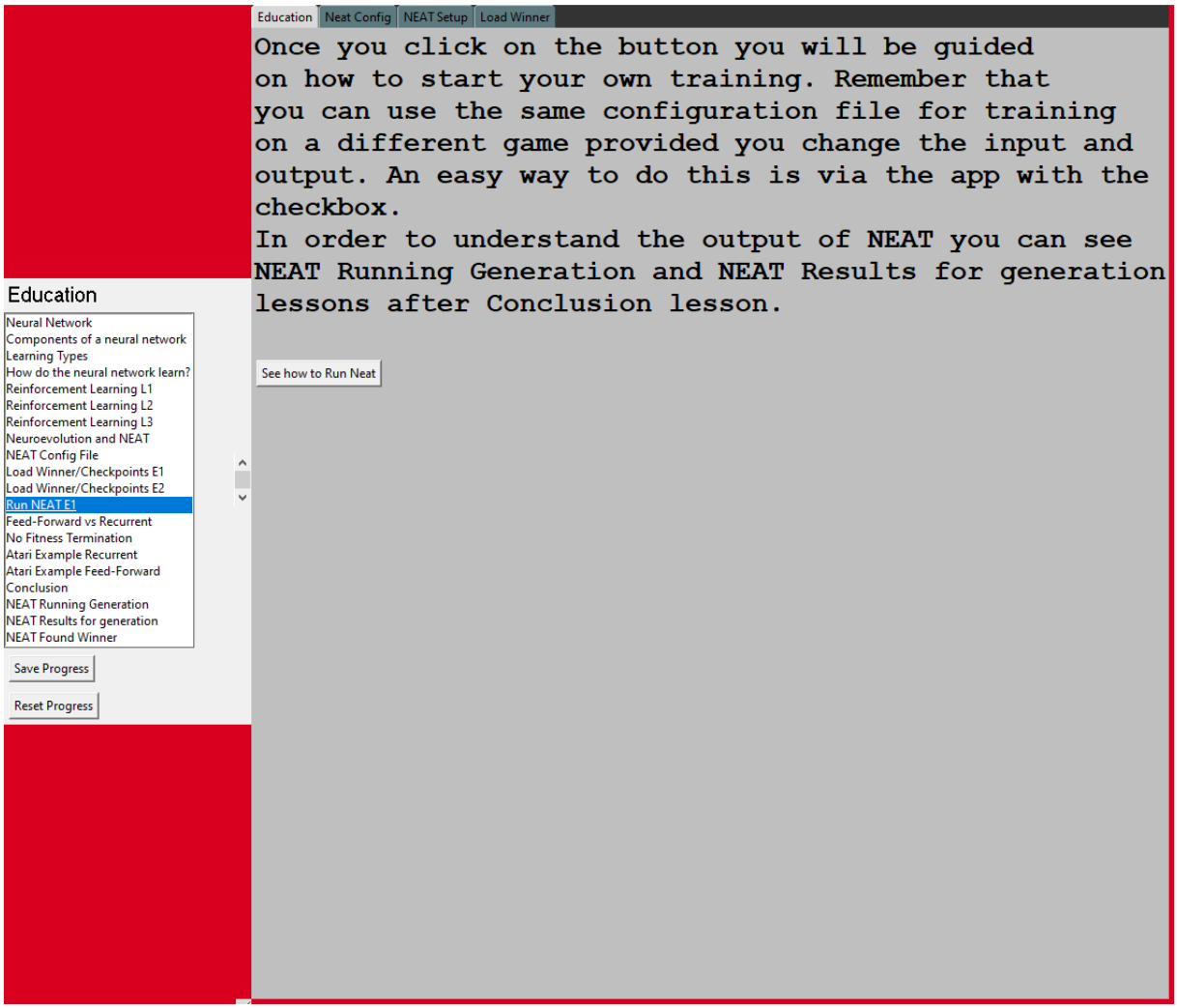


Figure 47 Example lesson with "Next" button

Upon pressing the button in these lessons, the tab focus will change and display the appropriate Sticky Note for the lesson.
The lessons must be taken sequentially in the current prototype, and in order to restart and start from the beginning, the user would have to close down the application and relaunch it. The program keeps a history of the progress for the passed lessons using a hidden value that changes when a button is clicked. The button is integrated only for Education purposes, and its sole purpose is to provide a guided education path. The button can be found in the Education tab, then either NEAT config, NEAT Setup or Load winner or even multiple, but it always starts from the Education tab and redirects to another and circles back to the Education tab next lesson.
The guided lessons that show the user how to use NEAT are developed with having in mind that the user may not know anything about NEAT and how to use it, so it explains the very basics and advances the difficulty thought the app getting into hidden features of NEAT that can be only discovered if the Configuration file and how to use NEAT is adequately understood.

4.3.8 Changes within libraries

A few changes within the libraries have been made to include additional functionalities or make the program available to be executed from another machine.

4.3.8.1 Atari_py changes

Problems such as the atari_roms were not found when launched from an executable file or overall from an IDE. The process of finding the directory atari_roms within the library was changed to detect where the script is running (source folder) and then merged atari_py/atari_roms with the path. The initial process was finding the directory of the script file, but this created issues when distributing to other systems as it could not find it as it was searching in the wrong location.
The problem was heavily seen when the program was built as an executable file. It launches from the Temp folder on the computer, and the search performed pointed to the incorrect directory.

4.3.8.2 NEAT changes

Checkpoints

The initial usage of the checkpoints of NEAT is to save the progress of the training so it can be resumed later. NEAT provides a specific command that allows NEAT to restore the checkpoint loaded. Checkpoint restoration includes population, config file, network type and winner file name; the user, in this case, would not need to specify anything as any other settings selected will not be considered. (See Fig. 40)
In table 11, it can be seen what the desired functionality is and how was it implemented. This has been implemented following User Story 7 from Load Winner.

Desired Functionality	How was it implemented?
Be able to see the progress of the neural network training at a particular generation to understand how the neural network improves over the generations.	The algorithm within the program for viewing the checkpoints uses the same analogy as starting the training but with the modification of checking how many genomes are left to view (if specified – this is vital as the user would view all the genomes in the generation if not specified) and modify the genome fitness to “Exit”. In the neat algorithm in population.py, where it loops throughout the genomes, if the genome fitness is empty, it will define it as “Exit”. In order for the algorithm not to crash with the modification - all of the math.util functions have an extra check if the value is not “None”; otherwise, it appends “0.0”. In reporting.py, the generation function receives the “Exit” fitness; a check is being performed if the fitness of a genome is equal to “Exit”, and it ends the algorithm. Finally, in population.py, in looping through genomes to see if the threshold is met, “AttributeError” or “TypeError” is generated to stop the algorithm. (See Fig. 48)
Save generation 0 if any further checkpoints are saved as if the user selects to save every ten generations it will save the 10th generation as checkpoint 0, then the 20th generation will be checkpoint two.	In the checkpoint.py file, a condition is added where the current generation is checked whether it is 0, and if it is, it creates a checkpoint. (See Fig. 49 – NEAT will save every five generations + generation 0)

Table 11 Neat Checkpoint Desired Functionality

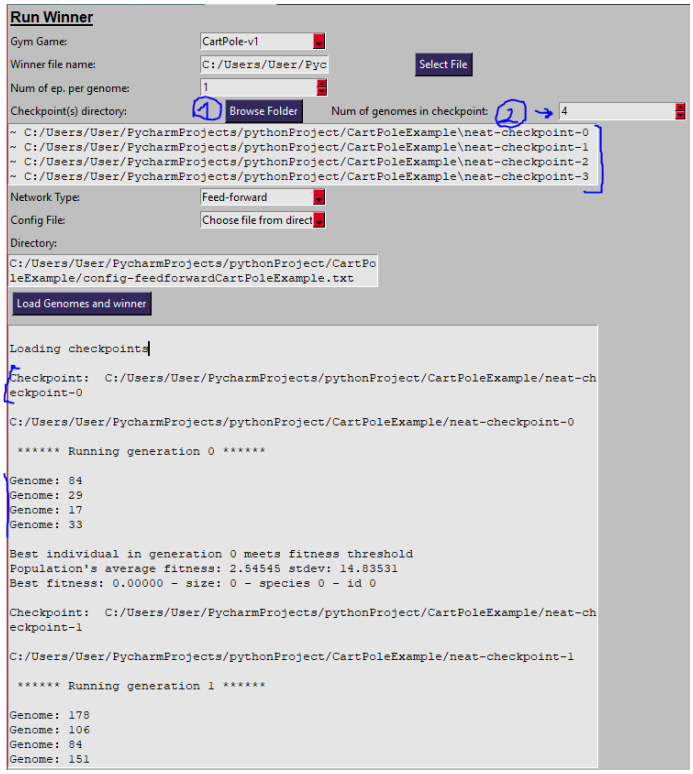


Figure 48 View Checkpoint generation progress

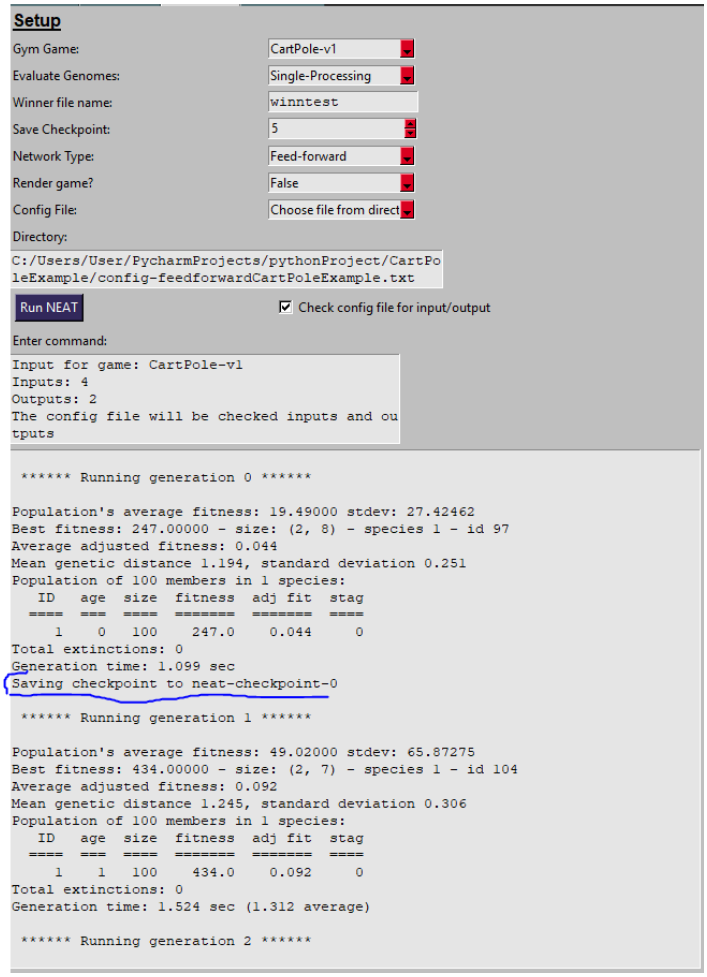


Figure 49 Neat-Checkpoint-0 Save

4.4 Problems with implementation

Problems during the program's implementation have been encountered, and for some – workarounds have been found. Others have been fixed to the point where it is possible.

4.4.1 Setting up Gym and Atari-py

Final versions that the implementation are: Python (3.7.0), Atari-Py (0.2.6) and Gym (0.18.0). The machine must have Visual Studio C++ dependencies installed (Visual Studio C++ Build Tools). All Gym, Python and Atari-py versions were tried until a working example were compiled without any errors and installing external things (I already had Visual Studio C++ Build Tools, so I did not realise until later that it is needed for Atari-py to run).
For more information, all pre-requirements and how to install them step by step have been included in https://github.com/venetsia/NEAT-Easy-To-Use-App-With-Gym/blob/master/README.md

4.4.1.1 Gym[11]

Numerous issues have occurred while trying to get gym Atari-py set up to work with Python overall. At first latest versions of Gym (latest version 0.20.0 in early stages in the development of the program (currently the latest version is 0.21.0)), Atari-py

[11] Gym Open AI - https://github.com/openai/gym

(latest version 0.3.0) and Python (latest version - 3.9.0 (currently the latest version is 3.10.0)) have been tried, but none of the games was launching as an array was going out of range after starting the game. Lowering the version of Gym to 0.20.0 (changed how it is implementing the Atari games) and then to 0.19.0 (issues with game versions) was tried, but issues occurred, including not being able to find the ROM files (Atari games). Following documentation on installing ROM files locally was not an option. Finally, installing Gym 0.18.0 allowed the simple game CartPole-v1 to launch. Checking on the repository for the supported Python versions: Python 3.7, 3.8 and 3.9 on Linux and macOS. It can be confirmed that Gym (without Atari-py games) works with 3.9 and 3.8.
An attempt to get Gym 0.21.0 working was made due to the many issues experienced when trying to get the Atari-py games working as the new Gym supports ALE-Py, which replaces the legacy version Atari-py (used in this project – depicted (no longer supported)). Documentation (Farebrother, 2021) regarding the release was followed to attempt to work and replace the legacy version. The new version, though, came with new issues as there is not enough documentation to understand which games are supported in the new version as in order to use a game with Gym, one would need to specify the version of the game. The new Gym version 0.21.0 uses v5 games. When this was tested in an isolated environment, issues with rendering the game were experienced in all rendering types (ram, RGB, grayscale). Even though the new version of Atari seems to provide more functionality, it could not be understood which games can be used as several games were tested with the -v5 extension and an error message was received that no such version exists. The program was left to use the legacy version of Atari-Py.

4.4.1.2 Atari-py[12] and Python

Trying to launch any Atari-py (0.3.0) game with Gym version 0.18.0 failed. After trying to understand why the ale_c.dll could not be found, a solution in StackOverflow was found where the suggestion was to download the dependency from a personal google drive. This approach was not taken due to security concerns. On the repository of Atari-py, the supported systems and versions for the library are Linux and Mac OS X with Python 3.5, 3.6, and 3.7, where Windows is not officially supported. Lowering the Python version to 3.7 still produced an error message, so the Atari-py version was also lowered until no error messages were found when trying to render the game. The version of Atari-py that did not cause any issues with any of the games like game version, ROM file cannot be found/loaded (See Atary_py changes section), ale_c.dll missing. No external downloading of files as needed, as suggested in questions in StackOverflow.
Later in the development of the program, when it was being tested on another computer, the error message “The specified module could not be found” or “gym.error.DependencyNotInstalled: No module named 'atari_py'”. Trying to map this specifically for the machine proved unsuccessful. The resolution for this issue is to have Microsoft Visual C++ Build Tools installed is a must for the Atari games.
The program is tested on Windows 10, and Windows 11 runs from the IDE PyCharm without any issues as long as the versions of the libraries are compatible and have a few pre-requirements met. Complete guidance on what needs to be done before running the program can be found on: https://github.com/venetsia/NEAT-Easy-To-Use-App-With-Gym/blob/master/README.md

[12] Atari-Py repository - https://github.com/openai/atari-py

4.4.2 Tkinter

The graphical user interface for this project was selected to be the build-in library - Tkinter as it is easy to use and learn.

4.4.2.1 Resizing Text Boxes

A limitation of the library was found when trying to fix the interface of the application where all text boxed on forms have to be the same width. After trying out a few different widths, it has been found out that the spin box, combo box and text field would always be different and could never look the same. The width of the text field (top) is 17, while the spin box (middle) is currently 19, and the combo box (bottom) is 20. The difference can be minimal but still visible. If the width for the boxes is the same, the difference is visible even more. The best combination, so they look similar (if not the same), is to have the spin box one number smaller than the combo box and have the text field three numbers down from the combo box. (See Fig. 50)

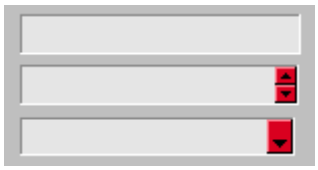


Figure 50 Text Field (top) vs Spin Box (middle) vs Combo box (bottom)

4.4.2.2 Displaying text (labels)

In order to display text in the tabs within the frames, the build-in widget label has to be used. Issues that were encountered in Tkinter labels can be viewed within table 12.

Issue	Why is it an issue?	Resolution
Getting the label widgets to be aligned the same way (on the same level)	When having different labels with different lengths of text, Tkinter default alignment looks odd and unstructured	Using ipadx[13] to move a widget to the left or right depending on the default position Tkinter gives. Changes had to be made to each label individually and tested each time the padding has been implemented to see changes and understand what further changes need to be done. The object-oriented approach might get complicated if attempted for the GUI
Changing the padding, so the size between the labels and boxes is the same (example Fig. 51)	As displayed in Fig. 40, the list widget for “Activation Func:” creates issues for the	Using ipady[14] did not seem to do anything, as changing the padding for one widget changed the padding for any widget in that row. A

[13] Tkinter ipadx – used in Tkinter library to change the padding vertically for widget in pixels
[14] Tkinter ipady - used in Tkinter library to change the padding horizontally for widget in pixels

	whole row due to the size height of the box	resolution for this issue has not been found.
Displaying a large amount of text with pictures (for example, in Education Mode lessons) – See Fig. 52	Displaying large amounts of texts proves to be quite tricky using labels: 1. Inserting the text on the same line without manually adding "\n"[15] will change the width of the whole program where it will display the text on one line and not adjust it manually. 2. Adding "\n" creates further issues as unexpected spaces between beginning and word can occur, so all must be configured carefully to align on the same level. 3. Fonts (Bhardwaj, n.d., p.) that Tkinter provides do not look nice when displayed in a large amount of text	This issue was resolved by not using Tkinter at all for this. The text and the pictures that need to be loaded are loaded via a picture. The picture is made with Adobe Photoshop and Paint 3D and pre-loaded with threads when the program is started. This fixed: the positioning of the text and images, having to create multiple parameters to store the different kinds of labels (if each label is on a separate row (due to pictures)), the font that Tkinter provides that does not look nice and friendly
Displaying a large amount of text with no pictures (for example, in Education Mode lessons) (See Fig. 53)		Adding "\n" to the end of each line while testing how much width would the text take and changing it afterwards. Modifications for each line must be made as on each line, "\n" must be specified. The spaces to indicate padding must also be added manually. (See Fig. 53)

Table 12 Issues with Tkinter Label Widget

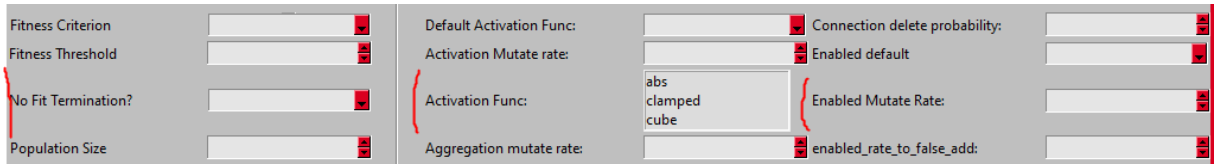


Figure 51 Issue with Padding vertically

[15] \n – new line

What is Artificial Intelligence?

When we think of artificial intelligence we can define it into four categories:

Systems that think like humans.	Systems that think rationally.
Systems that act like humans.	Systems that act rationally.

Figure 1 AI Definitions(Russell & Norvig, 1995, fig.1.1)

Strong AI
- when a machine truly understands what is happening
- there may even be emotions and creativity.
- for the most part, it is what we see in science fiction movies.
- this type of AI is also known as Artificial General Intelligence (AGI)

Weak AI
- a machine is pattern matching and usually focused on narrow tasks
- examples of this include Apple's Siri and Amazon's Alexa.

(Russell & Norvig, 1995, Chapter 1 Introduction; Taulli, 2019, Chapter 1 AI Foundations; Paul Mueller & Massaron, 2018)

Figure 52 Lesson from Education Mode

```
I hope you have learned some valuable information in the education
mode and you can now freely train a neural network and see the results
from it. If you would like to see the full guidence for the application
you can view it in the ReadMe file.

A few extra notes:
- in order to train your neural network so you reach a threshold, start
  with a lower threshold in order to view how the score increases in a
  game and what is the limit
- you can also use no_fitness_termination = True so you can run the
game for N generations to get an idea of how the game works
- you can reuse your config file given you change the input and output
  depending on the game you have chosen
- the games provided in the dropbox are tested games so any other game
  you try to input will not work
```

Figure 53 Lesson from Education Mode constructed with only one label

4.4.2.3 Scrollbar

The scrollbar in the Tkinter library does not implement well as it does not indicate if any lessons are left in the list as the scroll does not indicate it has any white space. (See Fig. 53)
Guidance on implementing the scroll bar has been followed, but it does not change no matter how it is integrated.

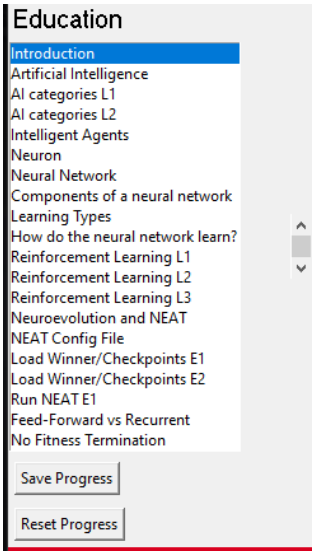


Figure 54 Scrollbar for Education Mode

4.4.2.4 Progress bar

The progress bar for this application was not integrated due to Tkinter not providing a nice-looking option.

4.4.3 Rendering Games

The Gym Library uses the Pyglet library for rendering the games.[16]
Initially, the application was developed single-threaded, but the graphical user interface was unresponsive until the algorithm finished running due to NEAT running on the main thread. This issue blocked some user stories from being implemented (User story 4) from Neat Setup and 8) From Load Winner). For this reason, a thread pool was created in order to submit the task of executing the algorithm on another thread separate from the main thread (where the graphical user interface is). The program was initially built with two threads – one for training the network and one for loading it. This created issues where if the rendering starts on one thread, the rendering will not work on the other thread. The reason for this has been found after extensive research that the Pyglet library cannot be imported within multiple threads, and once launched from a thread, and it cannot be transferred to another thread. This issue was fixed with having only one thread in the thread pool waiting to execute a task, so if the user wants to train a neural network and see the winner, the thread would have to finish whatever job it is executing and become available again, and it will then execute the other task. This ensures that the rendering of the games is happening only on the thread in the thread pool. Further tries to fix this issue have been attempted, so contacting a lecturer for concurrent and parallel systems ensured me that rendering libraries should typically operate in the main thread as things can get weird when one runs rending/windowing code from a thread that is not the main thread. Other rendering issues have not been seen once rendering from the thread in the thread pool has been implemented.

[16] Gym using piglet to render games - https://github.com/openai/gym/blob/master/gym/envs/classic_control/rendering.py

4.4.4 Neat

4.4.4.1 NEAT Config

Using a non-compatible config file version can create issues when running NEAT as it tries to match the Header Sections and then the parameters within. Between NEAT v0.92 to NEAT 0.9, there may be no reason between the Header Sections, but some parameters might be missing or have a different name, so it is vital to have a correct config file for NEAT v0.92 for this program.
The differences have been examined from personal experience and issues using a configuration file and looking at older versions of NEAT in the official NEAT-Python repository Tags section where the zip file was downloaded. The configuration file was examined or tested within the program.[17]

NEAT Version	Header Sections
NEAT v0.92	[NEAT], [DefaultGenome], [DefaultSpeciesSet], [DefaultStagnation], [DefaultReproduction]
NEAT v0.91	[NEAT], [DefaultGenome], [DefaultSpeciesSet], [DefaultStagnation], [DefaultReproduction]
NEAT v0.9	[NEAT], [DefaultGenome], [DefaultSpeciesSet], [DefaultStagnation], [DefaultReproduction]
NEAT v0.7	[Types], [phenotype], [genetic], [genotype compatibility], [DefaultStagnation], [DefaultReproduction]
NEAT v0.6	[phenotype], [genetic], [genotype compatibility], [species]
NEAT v0.5	[phenotype], [genetic], [genotype compatibility], [species]

Table 13 Differences between NEAT Config

4.4.4.2 Multi-Processing

Initially, Neat is meant to run on a single processor and multiprocessors. The advantage when running NEAT on multiprocessors is that the speed for its execution is significantly shorter.
The program could not be built to execute NEAT on multiprocessors due to the graphical user interface. When attempting to implement the multiprocessing feature of NEAT, it does not give any errors when compiling the application. However, when the algorithm is launched with a multiprocessing option, it starts the graphical user interface on multiple processors rather than just executing the algorithm (train the neural network on multiple processors). The multiprocessing functionality was tested, and it works without any issues without the graphical user interface, but when implemented, it starts executing the app on multiprocessors. The functionality was removed because of this issue. Contacting a lecturer[18] for concurrent and parallel systems, I was made aware that threads are limited in python and in general, I should use threads only to avoid input/output latency rather than utilizing multiple

[17] NEAT-Python repository Tags - https://github.com/CodeReclaimers/neat-python/tags
[18] Lecturer of Concurrent and Parallel Systems Babis Koniaris b.koniaris@napier.ac.uk (See conversation in Appendix 4 11.5)

cores (Ajitsaria, 2012). Threads in this program are used to avoid input/output latency where the NEAT algorithm is launched on a thread to avoid the graphical user interface becoming unresponsive when the user tries to do anything.

4.4.5 Creating an executable file

One of the most significant issues with the application is creating an executable file that the user can use to launch the application without needing to clone or download the repository, meet all pre-requirements, and launch from Pycharm after installing requirements and setting up Virtual Environment.
Various libraries displayed in Table 14 that build an executable file from the source code were tested. All builds needed to happen within PyCharm from the Terminal of the venv[19], otherwise would fail for all libraries (would not even build a .exe)

Library	Issues
Pyinstaller	Build an executable file but need specific libraries' mapping and map any dependencies the app is using. The application did launch on the computer. It was built on and on another machine, but testing the .exe on another machine failed to: find the atari_py module, missing a DLL file, imports not mapped correctly. Libraries are mapped in options.txt, where it specifies where to look for libraries and DLLs, but after trying to fix the issues, Pyinstaller was broken, and it could not build even a non-working executable file. Pyinstaller may work in the future, but it is needed to figure out how to map the C++ Build Tools needed for the atari games. See Fig. 55, 56 and 57 for experienced Atari-py error messages while executing on another machine.
Auto-py-to-exe	The application did not launch, and an error message was displayed (incorrect mapping of libraries)
cx_Freeze	The application does not launch on another machine and displays an error message. On machine build, it has issues loading Atari games.
Nuitka	Issues getting Tkinter to compile with Nuikta
py2exe	The application does not launch on another machine and displays an error message. On machine build, it has issues loading Atari games.
bbfreeze	No support for Python 3
py2app	Only supported for Mac OS
PyRun	Supports Linux, FreeBSD and Mac OS X.

Table 14 Python to Exe libraries tested

[19] Venv - The venv component supports creating lightweight “virtual environments” with their site directories, optionally isolated from system site directories. A virtual environment is a way to have several, parallel instances of the Python interpreter, each with unique package sets and several configurations. Each virtual environment contains a distinct copy of the Python interpreter, including copies of its support utilities. (Yegulalp, 2021)

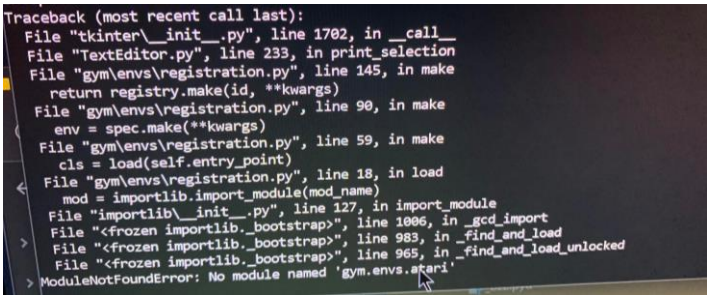


Figure 55 Launching .exe from PyInstaller Error Message Example 1

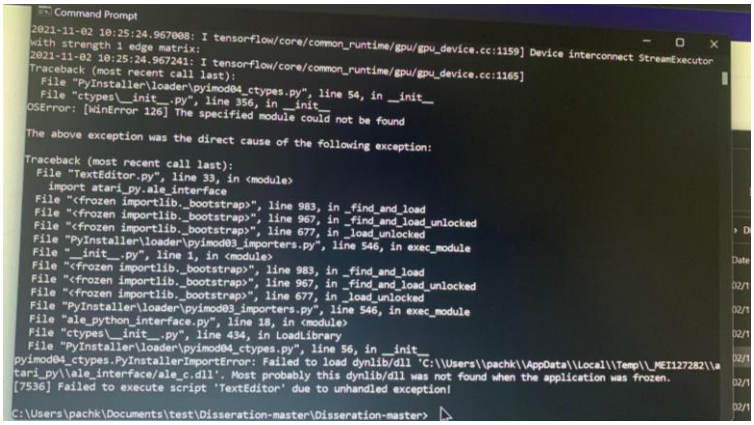


Figure 56 Launching .exe from PyInstaller Error Message Example 2

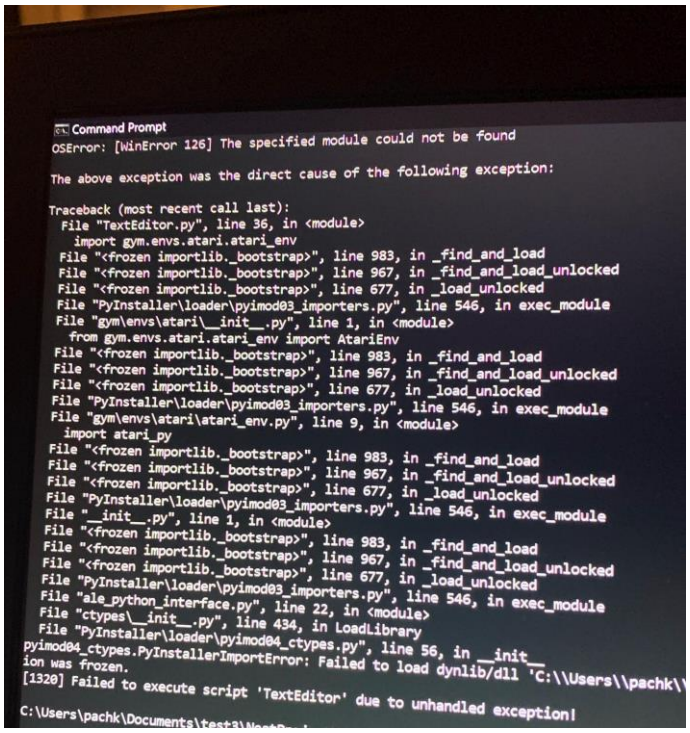


Figure 57 Launching .exe from PyInstaller Error Message Example 3

4.5 Programming Principles

The software was developed following the five general programming principles (Saikat Dutt & Geetha Chandramouli Subramanian, 2015, Chapter Software coding):

- Validity – the program gives correct results

- Consistency – the program does what it intends to do
- Maintainability – the program is easily changeable
- Readability – the program is easy to read and easy to maintain
- Usability – the program has a usability purpose of allowing a user to use NEAT without any coding necessary effortlessly

4.5.1 Coding Conventions

4.5.1.1 Naming Convention

Python's Style Guide was followed when writing the program's code to increase readability and consistency through the Python project and within the project itself (van Rossum & Coghlan, 2001). All of the variables, functions, objects and classes are named appropriately in order if another programmer reads the code, they would be able to understand its usage and purpose. All namings and comments are written in English.

4.5.1.2 Programming Principles and Rule of Thumb

Following various programming principles and rules of thumb is a practice senior software engineers follow, so for writing this program, the same has been done (Saikat Dutt & Geetha Chandramouli Subramanian, 2015, para. 8.4.2 Programming Principles and Rule of Thumb). See table 14 for how the best practices were followed.

Best practices	How is it followed:
Appropriate and sufficient data type for a variable	The data types for the variables are determined in run time due to the program being written in Python, which may affect the speed.
Avoid declaring all variables as a global variable	Global variables have been avoided as much as possible as they are generally bad practices in programming. Some variables like education mode lesson states and pre-loaded images are global variables to allow easier use in different classes and files.
Specific name for variables and methods	variable, method and other things are named according to the user, so it is easily readable and maintainable
variables and methods for the only purpose	each variable, method, and class have specific as multipurpose variables, and methods tend to be confusing
While using a database connection, create the connection object as late as possible and release it as early as possible.	No database used
Avoid using database connections using specific users' credentials.	
Avoid using the session variable in ASP. Session variables are stored in local machines.	

Avoid using forced data conversion (converting integer to long or float to be avoided).	Conversion is rarely used in the program. Its usage is for converting certain variables to meet requirements for text box, path directories, values obtained from text boxes
Release object references wherever not being used to release memory usage for some other object.	Python automatically frees all objects that are not referenced anymore (Golubin, 2020)

Table 15 Following best programming principles, Source: (Adapted from Saikat Dutt & Geetha Chandramouli Subramanian, 2015, para. 8.4.2 Programming Principles and Rule of Thumb)

4.5.1.3 Comments

Comments can be found through the program to ensure that if another developer tries to understand the process of the program and make modifications, everything is clear as comments serve no purpose for the compilation of the program but are used to understand the code itself and improve readability.

4.5.1.4 White Space Best Practices

Blank lines and white spaces improve the code's readability depending on whether it is a method (single blank line) or a top-level function/ class definition (two blank lines). They are implemented following Rossum & Coghlan (2001, sec. Blank Lines) official python guidance and following advice from Saikat Dutt & Geetha Chandramouli Subramanian book (2015, sec. 8.4.4 White Space Best Practices).

4.6 Testing

Testing the software is used to identify any issues in the program and provides information about the system's quality. This gives enough data to fix errors and create a working version.
Software testing evaluates the software product if it meets the required conditions and needs test cases, test data, and test results to compare the results with the excepted output whether it passes or not, and that produces test reports.

4.6.1 Test Plan

Due to the project being developed in an Agile way, the testing for each task is done in the same sprint to confirm their correctness and performance. After a task has been completed, testing is done as the agile way dictated that the prototype should be working on every sprint.
Testing is applied to each of the features the application has.

4.6.2 Test Methods

All tests have been performed by the program developer, and where another person needs no coding experience. The program has been regularly demoted to a supervisor on the weekly sprint meetings checking in with progress and testing app and identifying issues that need to be fixed.

4.6.2.1 Accessibility Testing

This testing ensures the application is working for users with or without disabilities. Throughout the application development, the colour pattern was tested with a user (informally) to ensure everything seemed clear, even for colour blind or partly colour

blind. Unfortunately, there is no audio within the application, so it would not make any issues for people with hearing disabilities.

4.6.2.2 Vulnerability Testing

Vulnerability testing has been performed on the system due to the dependencies it is using. Some dependencies that were being used had a security vulnerability, so they were updated to the most up to date version where no current security vulnerability has been performed. The program itself does not have any vulnerabilities as it does not store sensitive data.

4.6.2.3 Interface testing

Performed numerous times on one screen and numerous screens while having lots of applications open, and switching between the applications and screens does not make an issue. Furthermore, the function for NEAT's multiprocessing is not enabled, so the program does not take much CPU time as it trains agents one by one, whereas the multiprocessing trains agents in parallel.

4.6.2.4 Compatibility testing

In the early stages of the program, compatibility testing has been done due to the integration of the education mode to confirm that it works as the software was designed to switch to the next tab or next item on the list using image recognition. That created an issue with the integration on Windows 11, where the program was initially designed on Windows 10, so the education mode's whole functionality was changed to work on other operating systems using the built-in library of Tkinter. The final prototype was again tested on Windows 11, confirming that all of the functionality's work given the required dependencies are installed. The prototype is not tested on Linux or macOS but given is it distributed via code, it should cause no issues.

4.6.2.5 Performance testing

Performance testing was performed throughout the whole development of the program during each sprint when the task was finished. Issues in the early development of the program:

- When the NEAT algorithm was launched, the graphical user interface (GUI) was not responding until the algorithm finished.
    - Fix: integrate a thread pool to enable NEAT to run on another thread from the main thread to keep the GUI responsive.
- In education mode, loading a lesson caused a delay as lessons were loaded each time the lesson was opened.
    - Fix: pre-loading the images on a separate thread from the main one so that there is no delay or lag when the user starts to open the lessons.

4.6.2.6 Usability testing

Usability testing has been performed since the development of the program due to the nature of the evolutionary algorithm being quite hard to understand. The

Education mode and complete guidance make the application's usability more straightforward and more understandable.
The program's usability was not successful using only an executable file because when it was run on a different computer, it failed to run the Atari games. After a considerate amount of time trying to test why that is not happening, even with cloning the repository on a different machine, it was found out that the user would need to install Visual Studio C++ Build tools to run the Atari games successfully.
Furthermore, on the mapping for the Atari games, it was unsuccessful to map these dependencies to get the executable file working, so guidance on what needs to be installed before running the program has been supplied.
Another issue with compatibility with running the code on another device has been found with the path locations of specific files within education as the examples on how to use NEAT have pre-loaded examples where the parameters are filled out, so the path locations were initially wrong, but that was fixed using: os.getcwd()[20].

4.6.2.7 Functionality Testing

Open AI Gym Games

The Open AI gym games were all tested before including them in the program. Due to various reasons, some of the games in the Atari library were not included.
Reasons for not including the games include but are not limited to:

- Breakout - freezes
- BipedalWalker-v2 – not launching, an error message appears
- AirRaid-v0 - image is going black once in a while
- BankHeist-v0 - player cannot move, appears to be stuck, and the game does not restart after a specific time or die
- BeamRider-v0 - nothing happens
- Bowling-v0 - nothing happens (there is no timeout or a way to die)
- ChopperCommand-v0 – freezes
- CrazyClimber-v0 - not broken, but points decrease from 10000, so it takes a while for the game to reset, so the agent might not learn for quite a while, given multiprocessing for Neat is not included
- DoubleDunk-v0 – freezes
- ElevatorAction-v0 – the man is stuck in the air
- Enduro-v0 - no fitness being produced
- FishingDerby-v0 - takes much time to finish training on a single genome
- Gopher-v0 - freezes on second Level or Second life
- Gravitar-v0 - after 15 generations still at 0 fitness (I do not understand the game)
- IceHockey-v0 - takes too long
- Jamesbond-v0 - takes a while to train as the player has four lives
- Krull-v0 - a little too complicated, so on Single-Core, it will take quite a while
- MontezumaRevenge-v0 - no timeout, on second genome agent started jumping, and that is all so it will never die
- NameThisGame-v0 - nothing is happening (the agent does not die, and octopus should be spreading his tentacles to kill the agent) - I think it does not know how to start the game as in demo in GYM Open AI it has a timeout, and stuff is happening

[20] os.getcwd() - returns the current working directory

- Pitfall-v0 - timeout is 20 minutes, so the agent will not do anything for 20 min at the beginning (not suitable for single processing)
- PrivateEye-v0 - timeout is 3 minutes, and the agent does not appear to die when being hit (maybe decrease points if he has any)
- Qbert-v0 - timeout is long, and no progress
- Riverraid-v0 - game froze after genome 1
- RoadRunner-v0 - does not do anything after a few generations
- Robotank-v0 - time out takes ages
- Seaquest-v0 - enemies cannot reach the top of the ocean, and the agent does not go below water, so it will never die (not sure how long is a timeout)
- Skiing-v0 - agent found a way to break the game, froze on the right side of the screen, and the next genome, it froze on the left side of the screen
- Solaris-v0 - nothing happens as the agent is not taking off with the spaceship (not sure how long is a timeout) - maybe the game is broken
- Tennis-v0 - will never serve the ball as it does not know how (not sure how long is a timeout)
- Tutankham-v0 - ENEMY will never start moving as the agent does not start moving if not trained
- Venture-v0 - agent does not move at all, long timeout if not killed
- VideoPinball-v0 – freezes
- WizardOfWor-v0 - may cause issues as wizards have to get out of a place to start killing
- YarsRevenge-v0 - freeze after death
- Zaxxon-v0 - takes too long to execute a single genome

Neat Form

All neat form fields have been tested if they get imported into the config file, that would be saved, so it is used for the training.

- Issues appeared with species_elitism and elitism
    - Fix: special conditions were integrated for the parameters.

All values inserted for the parameters have been tested, and validation has been included following the official Neat-Python guidance[21].

Education Mode

The functionality of education mode has been tested, and redirection when clicking on the button Next has been fixed to use redirection using code as initially it worked with image processing which caused compatibility issues.
All images are pre-loading when the application is loading, so no further delay is experienced, and images are showing correctly in lessons.

Neat Setup

The games included in the list are working games that would not cause any issues while training the agent to play the game.
Validation for all fields has been included, not allowing the user to enter anything that would break the NEAT algorithm. Furthermore, if any fields are empty but required, it will not let the user run the algorithm.
The parameters submitted in the form and config file are valid when running the NEAT algorithm with a rendered image or without works without any issues.

[21] NEAT-Python guidance - https://neat-python.readthedocs.io/en/latest/config_file.html

Load Winner

All the functionalities within Load Winner have been tested extensively.
Initial issues occurred with:

- loading the checkpoints where the program would not recognize that there is only one checkpoint to load.
    - Fix: within code using if and else statements for the program to recognise whether only one, multiple, or none checkpoint/s needs to be loaded.
- Mapping the path directory to the config file created issues when providing a full path
    - Fix: within code to convert the path string into a raw string and "/" to "\\"

Dark Mode

The dark mode functionality has been tested throughout development. Different colour patterns have been tested, and examples of Dark Mode from the specific application have been used as guidance (PyCharm, Microsoft Word, Evernote).

Switching between Education mode to Normal Mode

Switching between the two modes does not create any issues, does not delete any of the filled-in variables and does not interfere with the NEAT algorithm running.

4.6.3 Test Cases

The test cases are a set of actions executed to verify a particular feature or functionality whether or not it is giving, returning or accepting the correct value. This has been implemented via test-driven development, where the software requirements are converted into test cases within the validation of the values or functions before the software is fully developed. Test cases are developed in order to ensure that the data meets the expected results. Test cases were not written in unit tests but instead implemented into the validation process of the fields and functions.
Test cases are implemented within each function and for each data field that needs extra checks to prevent an incorrect input from the user from breaking the application. The whole process for the validation is wrapped around a try and except clause where each test case may have a further try and except clauses depending on the needs.

4.6.4 Test Output

The test outputs are presented to show what has been done if the expected output is not met. The columns for all of the tables are as follows: "Input" (the input we are testing against), "Expected" (shows if the test has to pass or fail), "Passed" (shows whether or the returned output is the same as the expected output), "Issues" (shows what is the issue if the returned value is not the expected value), "Resolution" (how the issue got resolved, so the expected value is returned").
The rows highlighted in the red show when the expected value is not returned and is fixed.
More test outputs are not covered in this report, such as when before/while executing a function, but the test Output for Neat Config and Load Winner can be found in Appendix 4.

4.6.4.1 Neat Setup Validation

The below table (See Table 16) shows the validation process for the whole Neat Setup tab for the form when the user inputs a value.

Input	Expected	Passed	Issues	Resolution
Game_selection = “”	Fail	No	Value cannot be empty	Define a list of accepted games and see if value is one of the values in list
Game_selection = a	Fail	Yes		
Game_selection = 1	Fail	Yes		
Game_selection = SpaceInvaders-v0	Pass	Yes		
Game_selection = Breakout-v0	Fail	Yes		
Game_evaluation = “”	Fail	No	Value cannot be empty	Define accepted values and check if value is one of the ones on list
Game_evaluation = a	Fail	Yes		
Game_evaluation = 1	Fail	Yes		
Game_evaluation = Single-Processing	Pass	Yes		
Game_evaluation = Multi-Processing	Fail	Yes		
Winnerfilename = “”	Pass	Yes		
Winnerfilename = a	Pass	Yes		
Winnerfilename = a!	Fail	No	Special characters are not allowed in order not to make issues when getting path to winner file	On entering a character in the textbox, the input is validated and deletes any special characters
Winnerfilename = a1	Pass	Yes		
Winnerfilename = a 1	Fail	No	There should be no spaces	On entering a character in the textbox, the input is

				validated and deletes spaces
Game_checkpoint = “”	Fail	No	Value should not be empty	Colour label red to signal user
Game_checkpoint = a	Fail	No	Value should not be of alphabetical characters	Check if value contains any alphabetical characters
Game_checkpoint = 1	Pass	Yes		
Game_checkpoint = -1	Fail	No	Value should not be negative	Replace “-“with “”
Game_checkpoint = 0	Pass	Yes		
Game_checkpoint = 0.0	Fail	No	Value should be of type int not float	Convert value to int - (int (text))
Game_checkpoint = 1.0	Pass	Yes		
Network_type = “”	Fail	No	Value should not be empty	Colour label red
Network_type = a	Fail	No	Value should be of a certain list	define an allowed list and check if value is one of them
Network_type = Feed-forward	Pass	Yes		
Network_type = Reccurrent	Pass	Yes		
Network_type = 1	Fail	Yes		
Render_game = “”	Fail	No	Value should not be empty	Check if value is empty and colour label red if it is
Render_game = a	Fail	No	Value should be of a certain list	Define an allowed list and check if value is one of them
Render_game = True	Pass	Yes		
Render_game = False	Pass	Yes		
Render_game = 1	Fail	Yes		
Num_generations = 0	Fail	No	Value cannot be zero	Perform a check if value is int and if not set to 1
Num_generations = a	Fail	Yes		
Num_generations = 1	Pass	Yes		
Num_generations = 1.0	Fail	No	Value cannot be of type float	Check if value is valid float and convert to int

Num_generations = 1.0a	Fail	Yes		
Num_generations = 0.0	Fail	No	Value is not valid float	Check if value is 0 or 0.0 and set to 1
Num_generations = -1	Fail	No	No negative values	Replace “-“ with “”
runs_per_network = 0	Fail	No	Value cannot be zero	Perform a check if value is int and if not set to 1
runs_per_network = a	Fail	Yes		
runs_per_network = 1	Pass	Yes		
runs_per_network = 1.0	Fail	No	Value cannot be of type float	Check if value is valid float and convert to int
runs_per_network = 1.0a	Fail	Yes		
runs_per_network = 0.0	Fail	No	Value is not valid float or zero	Check if value is 0 or 0.0 and set to 1
directory_value = <any manual input>	Fail	No	User should not be able to modify path	bind('<Key>',lambda e: 'break') – Bind widget to any input and prevent
directory_value = <chosen path from prompt>	Pass	No	Widget prevents any inputs	Insert text into widget by enabling the widget and then locking it

Table 16 Tests Output for Neat Setup

4.6.5 Preliminary experimentation

During the functionality testing of Gym games, an exciting discovery was found for Space Invaders where the system would perform better with Feed-Forward and not Recurrent Neural Networks even though the same configuration file and specifications were set. This may be because, in space invaders, there is no advantage to having time-series data.

4.7 Evolution

4.7.1 Maintenance

Software maintenance is one of the core aspects of software engineering. Software maintenance is the process of modifying the production system after the delivery to correct the faults, improve performance, and adapt to the changing environment (Saikat Dutt & Geetha Chandramouli Subramanian, 2015, Chapter Software Maintenance).
Maintenance of the application takes place when the application is released to the public.

4.7.1.1 Adaptive Maintenance

Changing the system based on environmental adjustments is called adaptive maintenance. This maintenance is performed when the system has no issues, but the environments are being used change. No changes in functionality occur during this maintenance (Saikat Dutt & Geetha Chandramouli Subramanian, 2015, Section Adaptive Maintenance).

Adaptive maintenance for the application may occur if:

- Security vulnerabilities are found within libraries used
    - Versions of the libraries need to be updated to meet requirements, and some changes may need to be done in the code if the libraries usage is changed, such as single commands
- Meet Operating System requirements
    - Currently, the system is tested on Windows 10 and Windows 11, but for the Atari games to work, the Windows 10 SDK (even on Windows 11) needs to be installed from C++ Build Tools. If newer versions of Windows do not provide the option to install the SDK, the Atari games will not work, and an alternative route to get them working needs to be found, such as running the program on an isolated system such as Docker or running the application from a Virtual Machine using an operating system that can have Windows 10 SDK installed.

4.7.1.2 Corrective Maintenance

According to Saikat Dutt & Geetha Chandramouli Subramanian (2015, Section Adaptive Maintenance), corrective maintenance consists of identifying, isolating, and repairing any faults (errors) discovered within the system can be restored and operational.

There are two types of corrective maintenance:

- Immediate corrective maintenance – work begins instantly after the crash, and it is critical to the system
- Deferred corrective maintenance - postponed in conformance to stakeholders

The issues identified can be user-based, coding (software-based) or hardware-based.

The system has been tested extensively to prevent the need for immediate corrective maintenance, so no issues requiring this kind of maintenance have been identified. The need for such maintenance should be user-based faults where an exception is not handled correctly. Even though the system handles exceptions, an unexpected one can be found, as with any system that exists.

Deferred corrective maintenance may occur if fixing the issue has to be delayed due to a higher priority error that needs to be fixed, lack of needed skills for fixing the issue or no planned strategy to fix the issue (Deferred Maintenance How-Tos and Resources | Limble CMMS, 2019; Comprehensive Guide To Corrective Maintenance | Limble CMMS, 2019).

4.7.1.3 Perfective Maintenance

This type of maintenance includes upgrading the hardware and software due to adapting changes and user requirements.

Indeed, upgrading the application to use the latest Gym and Atari-py would be the first prioritised task of perfective maintenance if another with higher priority does not appear.

If an executable file is to be created, it would need to be upgraded for each new operating system that comes out as Python libraries that create executable files need

the build to happen on the machine that the executable would be used. See 4.4.5 for issues with building a .exe file. (Saikat Dutt & Geetha Chandramouli Subramanian, 2015, Section Perfective Maintenance)

4.7.1.4 Preventive Maintenance

This type of maintenance aims to prevent any outages or errors that the system may experience. Identifying components of the system that might fail for any reason possible is critical. Through the development of the program, testing each feature as it is implemented and thinking of ways to create an error message or a severe issue for the program that might freeze or break it have been considered and taken care of caution. Preventative Maintenance still should be considered to decrease the possibility of immediate corrective maintenance. (Saikat Dutt & Geetha Chandramouli Subramanian, 2015, Section Preventative Maintenance)

4.7.2 Factors to reduce maintenance cost

Functional-oriented designed files could be transformed into object-oriented designs to keep maintenance easier.
While the system is easy to maintain, having external people know the system's code would be trivial as they would have to understand how the whole program works to maintain it.
Implementing the program to use the new Atari library might be beneficial as the one used in this prototype is a legacy version that developers no longer support.

4.7.3 Maintenance prediction

For the maintenance of the system, developers should always consider:

- Expensive parts to maintain: Maintaining the application should not be expensive, given that the developers understand how NEAT works and how it connects to the Gym library. Therefore, everything within the application is available for restructuring. Adding functions, removing or changing should be efficient and straightforward once a developer knows the system and the graphical user interface.
- Lifetime maintenance cost: usually, maintaining the program would be higher than the development as changing something might break something else, so further maintenance and troubleshooting would be required.
    - Some libraries might become depicted after time so that no external support will be available.
        - There were some security vulnerabilities on previous versions of the application, with some of the external libraries being used, so a newer version has been specified in the requirements.txt file.
        - The Atari-py library that the application uses is no longer supported as a newer version exists[22] (Farebrother, 2021) but switching to the newer working version of GYM would need reengineering the Run_winner.py and Neat_Single_Processing.py as the usage of the Gym library is different in the new version.
    - Translating the program to another language might create difficulties with libraries that are not supported in other languages or have different usage.

[22] Atari Learning Environment new version - https://brosa.ca/blog/ale-release-v0.7

    - Some libraries might stop working after time so reengineering the application has to take place for the application to continue to work.
- Part of the system that is most likely going to change:
    - Functional-Oriented Designed files would be changed to Object-Oriented Design
    - More lessons for the educational mode can be added
    - In education mode within examples, the process is likely to change as currently, once you click Next, you have to restart the program to start from Example 1
    - Support for the newer version of the Gym library might be added in a newer / different version of the app
    - Support for older and (if any) new versions of Neat – changing the version of Neat would mess up currently how the program executes as launching the algorithm, and the configuration file are different for older versions

4.7.4 Reengineering

The cost of reengineering an existing system would be less than developing new software. However, after the system has been maintained for a long time and its cost has increased over time, reengineering can refactor the system to make it better. The purpose of reengineering the application would be to improve performance or change the design of the application. However, the application is developed for educational and research purposes would depend on reengineering.
Factors that may enforce reengineering the program:

- Libraries used in the program no longer work
    - This would mean that the algorithm of the program where it uses the libraries needs to be changed. For example, the highest risk library is Gym and Atari-py due to Atari-py being depicted, and games included in the library may stop working overtime for external reasons out of the developer's scope of the prototype.
        - There is currently a newer version of Gym and Atari-py, but they do not work as well as the current version. The program would not need much changing as it would have the same analogy for training and loading algorithm if NEAT version still works but rather the version of the games used as currently the latest versions of games is v5 for new Atari-py library (Ale-py directly imported into Gym library)
    - The prototype uses the latest version of NEAT, but as time passes, more versions can be developed, and as discussed in 4.4.4, there is a considerable difference between the versions, so reengineering would have to take place for editing the config file and within training and loading winner. To this date, training a neural network and loading the winner has no difference between the versions, but the configuration files have different parameters.
- Improve performance on a guided walkthrough of the application where the user would not need to restart the program to view examples.
- Meet new requirements (since this is an educational application, not all requirements would be a reason for reengineering as the application currently does work and does teach the user basic knowledge on Artificial Intelligence

topics and gives the user the freedom to train an agent and see and compare the outcome)
    - Requirements from users to include Multi-processing NEAT evaluation
        - A reason for the multi-processing NEAT evaluation not working was not found
    - Support older or newer versions of NEAT and GYM

It is up to the current maintainers/ developers/ organisation to scrap the system or continue maintaining the existing one.

5 User evaluation

The program must be installed on the users' computer for the user evaluation if testing would be done with the survey. Otherwise, with an interview, I would show the program for my computer and give remote control to the user if they would like to play around with the application. All instructions for installing and completing documentation of the application can be found on the GitHub repository.[23]
User evaluation might be affected by the graphical user interface as Tkinter has many limitations, and it looks pretty technical using the library.
The ethics form from Appendix 4 11.3.1 was completed before creating the survey. Each user had to sign a Consent Form (Appendix 4 11.3.2 and 11.3.3) before continuing to instructions on accessing the application and survey questions.

5.1 Survey

The survey method has been used in order to evaluate the developed project. The questions are focused on the entire consumer experience covering educational mode and the overall feel of the application. Keeping the survey short ensures the user does not lose interest in completing it as the process should be painless and should not take too much time to complete, which maximises the response rate (Jones et al., 2013). There is only one open-ended question, as answering these questions requires more effort and time (Connor Desai & Reimers, 2019). Surveys help understand the unmet needs of your customers and tailor your service strategies to meet those demands

5.1.1 Questions

The questions used in the survey can be viewed in Appendix 4 (11.3.5), where Q1-Q4 use the Likert Scale to measure the feedback and Q5 is an open answer.
The application's primary purpose is to be educational and easy to use, so the questions focus on understanding if the aim has been met. The questions emphasise gathering information if the Education mode provides enough knowledge, so the user feels comfortable using the application afterwards and is brief and focused. The questions asked are chosen to gain feedback from users on Education mode, so specific changes can be made depending on the answers received. One question focuses on understanding if the application is easy to use because the main functions of NEAT are implemented accordingly and it would be beneficial to know if they are presented understandably. In contrast, for Education mode, it is vital to know if the lessons were sufficient enough for the user going through them.

5.1.2 Likert Scale

The Likert scale has been used as it is pretty standard (Losby & Wetmore, 2012; Deckelmann, 2019), easy to use, and users who have done surveys are familiar with it. The results are simple to understand and analyse what can be improved.

5.1.3 Methodology

The survey is hosted in NoviSurvey[24], which allows gathering data on and off-campus, beneficial as the survey is meant for students and people interested in AI.

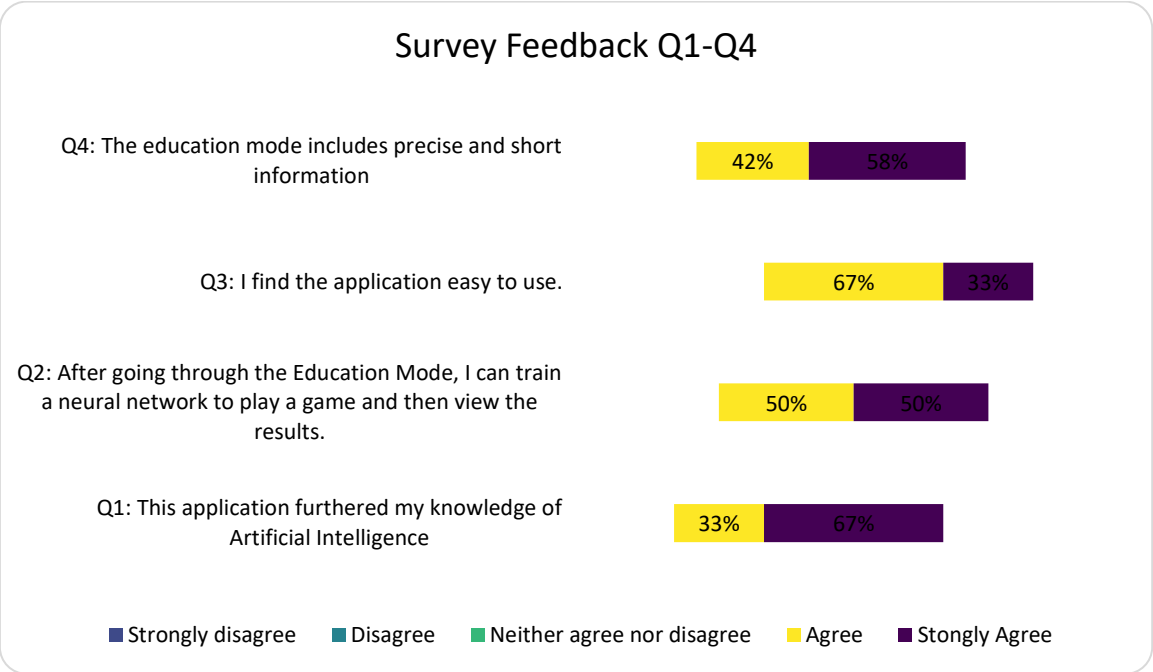
[23] GitHub Neat easy to use app - https://github.com/venetsia/NEAT-Easy-To-Use-App-With-Gym
[24] NoviSurvey - https://survey.napier.ac.uk/

The survey has been distributed via post/email/ message to:

- work colleagues who are interested in AI and found the project interesting
- shared on Moodle page with students currently studying the module Artificial Intelligence
- Student Groups and forums within social media
- Developed of NEAT algorithm
- Friends interested in AI

5.1.4 Results

Representing the twelve submitted survey results (See Appendix 4 11.3.6 for raw data) in a Diverging Stacked Bar Chart is the best way to visualize the Likert scale according to various sources (Bounthavong, 2019; 2018; Wexler, 2021; *How to Present Likert Scale Data?*, n.d.). The guidance that Bounthavong (2019) describes has been followed for creating the chart. The colours used in the chart are from the Viridis Colour Palette to be perceived by readers as the most common form of colour blindness. The Viridis Colour Palette contains four sequential colour scales (Neth, 2021, sec. D.4.2; Nuñez et al., 2018). Nuñez et al. (2018) compared the different colour palettes for different kinds of colour blindness, and it can be seen that the Viridis and Magma scales do better in covering a wide perceptual range.
It can be seen from the chart that the users either Agree or Strongly Agree with the sentences. For Q3, 80% of the users Agree with the statement, so there is room for improvement. Making the application straightforward for an algorithm to perform specific tasks requires the user to spend time understanding the algorithm itself and the methods the application can perform. For Q2, the Agree and Strongly Agree results have the same percentage, so including more examples within the application may shift more users to choose Strongly Agree when the application has been improved. Other questions mostly tilt to Strongly Agree, but there is always room for improvement as the users' background is unknown, so there is no indication if the user who answered has already some knowledge in AI.
Overall, the answers from the survey are positive, so a conclusion that the application is easy to use and improves the knowledge of users can be drawn as they can later use it to train a neural network and view the result.



Only one response for Q5 was received with feedback on guidance in GitHub while it was still being developed. Feedback was taken into account and integrated into guidance in GitHub as an overview of the application was missing as it is crucial information for people just finding the application by chance.

5.2 Interviews

The interview option has been given to users as an alternative option, specified in the repository and survey, for trying out the application. No users have been in touch to schedule a meeting, so no interviews were performed.

6 Conclusion

6.1 Achievements of aims

This project aims to build an easy-to-use graphical application accessible to researchers who do not have any coding experience and can still train a neural network using NEAT to control a game character in multiple Atari games. Building such an application would increase the accessibility of NEAT to people who are not comfortable with coding or are having issues. In order to achieve this, testing the NEAT had to be undertaken so I could understand the needs a user may have and develop all the features that NEAT provides and connect them with games.
Achievements of the program:

- Connect NEAT with Open AI Gym library in a graphical user interface
- Choose a game from a list of tested and working games (User Story 10 from Editor)
- Modify all options that can be specified before NEAT is launched (User Story 2 from Editor)
- Automatically get the input and outputs for a game integrated into config or view while editing the config file (User story 5 from Neat Setup (Train Neural Network))
- Load external config file; get an easier to understand the layout and edit it in program (User Story 1 from Editor)
- Load all the default parameters for NEAT in the config file with a button (User Story 3 from Editor)
- Easily edit the configuration file while validating the input and viewing the parameter's description. When validation is not passed, the user is alerted with a red label or the value is the default one if the parameter has one. (User Stories 7 and 8 from Editor
- Educate users on AI topics and how to use NEAT in the program (User Stories 3, 4, 5, 8 and 9 from Education mode)
- Train a neural network to play a game of the users' choice from a list (User stories 1, 2, 3 and 4 from NEAT Setup (Train Neural Network))
- Load the winner genome found from training and compare it with previous generations to see the progress (User Stories 7, 8 and 9 from Load Winner)
- See the progress of the training concurrently while NEAT is running in the program (User Stories 4 from NEAT Setup (Train Neural Network) and 8 from Load Winner)
- Users can use a few hidden features (hidden because configuration file has to be understood clearly in order to be used from code) that NEAT has – no_fitness_termination and restore from checkpoint

All User Stories have been implemented within the program but some initial scketch design features from Education mode are missing in the application.
Gathering information from different websites was done to analyse users' struggle when getting NEAT working with Gym library and testing NEAT with the GYM library to identify functions that can be implemented and understand users' requirements.

6.2 Critical Analysis

The project overall went well regardless of the issues faced during development as workarounds have been found or problems did not affect the completeness of the program, so it was left as it is.

The final product of this project meets all the requirements sets and provides the main functionalities of NEAT. However, even though issues have been experienced throughout development, a workaround for critical functionalities needed to be included.

Developing a graphical user interface with NEAT and Open AI Gym has been very rewarding as I have furthered my knowledge in reinforcement learning and feel confident in developing applications focusing on Artificial Intelligence. It has helped me to identify the practical difference between supervised, unsupervised and reinforcement learning.
I have also learned how to distribute an application with external libraries and changes to different machines without having an executable.
Even if all of the requirements of the application are met, connecting with more people who use NEAT and know how it works would be beneficial as a more detailed list of requirements to make NEAT accessible may have been provided as there may be some functionalities currently missing that I am not aware of myself.
During the development of education mode, the guided walkthrough examples were implemented with a lot of issues and workarounds due to the application, not the following architecture and due to the graphical user interface being in the main file where using specific values in different files proved to be quite challenging, so I would develop the structure of the application differently if I have to start the project again with the knowledge I have now. If the project should be started again, it would firstly be easier to develop the application, but the aim would be for developing the graphical user interface (Tkinter part) in an object-oriented manner in order to decrease the lines of code and make it reusable and having the graphical user interface not in the main file. I would include more questions in the survey to understand what people did and did not find easy to use if I were to repeat the project. Finally, education mode can be improved where more gamification features like levels, badges and cooperation and/or competition between users is integrated.

6.3 Future work

The project produced good results and met its aims for delivering an easy-to-use application, but if there were more time to develop the project, the following functionalities would be the priority of implementation, but due to the time to submission is close, these ideas are left for future work for the project:

- Integrate player vs computer play mode in games where possible
- Build different versions of the application to cover different versions for NEAT and Gym
- Implement visualizations for the found winner genome
- Include games from other libraries or environments
- Convert GUI part of the app into Object-Oriented design
- Use a different library for implementing GUI
- Include more tutorials for using NEAT, explaining in greater detail regarding the configuration file
- Try to implement multithreading NEAT
- Build the application as an executable file or upload it in Docker to be used from other computers without installing pre-requirements.
- Fix guided walkthrough in Education mode to not need a restart to view examples again.
- Implement other algorithms within the application to perform the same functionality as NEAT and compare results

- Include badges, progress bars or other self-elements or social elements from gamification. (Huang & Soman, 2013)

7 References

Admin, S. (n.d.). The Complete Guide to Choosing A Color Palette For Your eLearning Course [Blog]. Retrieved October 3, 2021, from https://www.shiftelearning.com/blog/the-complete-guide-to-color-combinations-in-elearning

Ajitsaria, A. (2012, 2021). *What Is the Python Global Interpreter Lock (GIL)? – Real Python*. https://realpython.com/python-gil/

Bellemare, M. G., Naddaf, Y., Veness, J., & Bowling, M. (2013). The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, *47*, 253–279.

Bellemare, M., Veness, J., & Bowling, M. (2012). Investigating Contingency Awareness Using Atari 2600 Games. *Proceedings of the AAAI Conference on Artificial Intelligence*, *26*(1), Article 1.

Bhardwaj, A. (n.d.). *python—List available font families in `tkinter`*. Stack Overflow. Retrieved March 23, 2021, from https://stackoverflow.com/questions/39614027/list-available-font-families-in-tkinter

Bounthavong, M. (2019, May 16). *Communicating data effectively with data visualization – Part 15 (Diverging Stacked Bar Chart for Likert scales)*. Mark Bounthavong. https://mbounthavong.com/blog/2019/5/16/communicating-data-effectively-with-data-visualization-part-15-divergent-stacked-bar-chart-for-likert-scales

Brockman, G. (2015). *OpenAI Gym: Gdb's algorithm on CartPole-v0*. https://gym.openai.com/evaluations/eval_w8MhbdYUT52bz7dKQrUvA

Brown, M. S. (2014). *Data Mining For Dummies*. For Dummies.

Cardamone, L., Loiacono, D., & Lanzi, P. L. (2009). Evolving competitive car controllers for racing games with Neuroevolution. *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, 1179–1186

Chukwuchekwa, J. (2011). *Comparing the performance of Backpropagation algorithm and Genetic Algorithm*.

Comprehensive Guide To Corrective Maintenance | Limble CMMS. (2019, April 25). Limble. https://limblecmms.com/blog/corrective-maintenance/

Connor Desai, S., & Reimers, S. (2019). Comparing the use of open and closed questions for Web-based measures of the continued-influence effect. *Behavior Research Methods*, *51*(3), 1426–1440. https://doi.org/10.3758/s13428-018-1066-z

Daume III, H. (2012). *A Course in Machine Learning*. http://ciml.info/dl/v0_9/ciml-v0_9-ch03.pdf

Deckelmann, D. (2019, June 11). What Are the Benefits of the Likert Scale? *LiveSurvey*. https://livecusurvey.com/likert-scale-benefits/

Deferred Maintenance How-Tos and Resources | Limble CMMS. (2019, October 10). Limble. https://limblecmms.com/blog/what-is-deferred-maintenance/

Ding, S., Li, H., Su, C., Yu, J., & Jin, F. (2013). Evolutionary artificial neural networks: A review. *Artificial Intelligence Review, 39*(3), 251–260.

Diuk, C., Cohen, A., & Littman, M. L. (2008). An object-oriented representation for efficient reinforcement learning. *Proceedings of the 25th International Conference on Machine Learning - ICML '08*, 240–247.

Farebrother, J. (2021, September 14). *The Arcade Learning Environment: Version 0.7*. https://brosa.ca/blog/ale-release-v0.7

Golubin, A. (2020, August 10). *Garbage collection in Python: Things you need to know*. Artem Golubin. https://rushter.com/blog/python-garbage-collector/

Hastings, E. J., Guha, R. K., & Stanley, K. O. (2009). Automatic Content Generation in the Galactic Arms Race Video Game. *IEEE Transactions on Computational Intelligence and AI in Games*, *1*(4), 245–263

Hausknecht, M., Lehman, J., Miikkulainen, R., & Stone, P. (2014). A Neuroevolution Approach to General Atari Game Playing. *IEEE Transactions on Computational Intelligence and AI in Games*, *6*(4), 355–366. 3

Huang, W. H.-Y., & Soman., D. (2013). *A Practitioner's Guide To Gamification Of Education*.

Hubans, R. (2020). *Grokking Artificial Intelligence Algorithms*. Manning Publications.

Iuhasz, G., Tirea, M., & Negru, V. (2012). Neural Network Predictions of Stock Price Fluctuations. *2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 505–512.

Jones, T., Baxter, M., & Khanduja, V. (2013). A quick guide to survey research. *Annals of The Royal College of Surgeons of England*, *95*(1), 5–7. https://doi.org/10.1308/003588413X13511609956372

Kinsley, H. (2017, March 13). *Python Programming Tutorials*. https://pythonprogramming.net/openai-cartpole-neural-network-example-machine-learning-tutorial/

Kumar, K., & Thakur, G. S. M. (2012). Advanced Applications of Neural Networks and Artificial Intelligence: A Review. *International Journal of Information Technology and Computer Science*, *4*(6), 57–68. 8

Levy, S. D. (2021). *Simondlevy/neat-gym* [Python]. https://github.com/simondlevy/neat-gym (Original work published 2020)

Liu, Y., Ghandar, A., & Theodoropoulos, G. (2020). Online NEAT for Credit Evaluation—A Dynamic Problem with Sequential Data.

Lorenz, A., & Whitby, M. C. (2006). Crossover promotion and prevention. *Biochemical Society Transactions*, *34*(Pt 4), 537–541.

Losby, J., & Wetmore, A. (2012). CDC Coffee Break: Using Likert Scales in Evaluation Survey Work. *Centers for Disease Control and Prevention*.

Lowell, J., Birger, K., & Grabkovsky, S. (2011). *Comparison of NEAT and HyperNEAT on a Strategic Decision-Making Problem*. 11.

Mirjalili, S. (2018). Chapter 4 Genetic Algorithm. In *Evolutionary Algorithms and Neural Networks: Theory and Applications (Studies in Computational Intelligence* (1st ed. 2019 ed., Vol. 780, pp. 43–55). Springer.

Moriarty, D. E., & Miikkulainen, R. (1996a). *Evolving Obstacle Avoidance Behavior in a Robot Arm*. 12.

Moriarty, D. E., & Mikkulainen, R. (1996b). Efficient Reinforcement Learning through Symbiotic Evolution. *Machine Learning*, *22*(1), 11–32.

Naddaf, Y. (2010, Spring). *Game-independent AI agents for playing Atari 2600 console games*. ERA.

Nederkoorn, C. (2021, April 9). Which GUI Framework is the best for Python coders? ActiveState. https://www.activestate.com/blog/top-10-python-gui-frameworks-compared/

Neth, H. (2021). *Data Science for Psychologists*. https://bookdown.org/hneth/ds4psy/

Nuñez, J. R., Anderton, C. R., & Renslow, R. S. (2018). Optimizing colormaps with consideration for color vision deficiency to enable accurate interpretation of scientific data. *PLOS ONE*, *13*(7), e0199239. https://doi.org/10.1371/journal.pone.0199239

Omelianenko, I. (2019). *Hands-On Neuroevolution with Python*. Packt Publishing. https://learning.oreilly.com/library/view/hands-on-neuroevolution-with/9781838824914

OpenAI. (n.d.). *Gym: A toolkit for developing and comparing reinforcement learning algorithms*. Retrieved November 20, 2021, from https://gym.openai.com

Open Source, G. (2018, 2021). *The Top 78 Machine Learning Openai Gym Open Source Projects on Github*. https://awesomeopensource.com/projects/machine-learning/openai-gym?categoryPage=12

Pintillie, D. (2010, May 19). *Beginners Guide to Using the Power of Color in Web Design*. Speckyboy Design Magazine. https://speckyboy.com/beginners-guide-to-using-the-power-of-color-in-web-design/

RADIGAN, D. (n.d.). Four agile ceremonies, demystified. Atlassian. Retrieved November 13, 2021, from https://www.atlassian.com/agile/scrum/ceremonies

Reeder, J., Miguez, R., Sparks, J., Georgiopoulos, M., & Anagnostopoulos, G. (2008). Interactively evolved modular neural networks for game agent control. *2008 IEEE Symposium On Computational Intelligence and Games*, 167–174.

REHKOPF, M. (n.d.). Sprints. Atlassian. Retrieved November 13, 2021, from https://www.atlassian.com/agile/scrum/sprints

Risi, S., & Stanley, K. O. (2012). An Enhanced Hypercube-Based Encoding for Evolving the Placement, Density, and Connectivity of Neurons. *Artificial Life*, *18*(4), 331–363.

Risi, S., & Togelius, J. (2015). Neuroevolution in Games: State of the Art and Open Challenges.

Rocheleau, J. (2020, March 9). *Basics Behind Color Theory for Web Designer*. Hongkiat. https://www.hongkiat.com/blog/basics-behind-color-theory-for-web-designer/

Russell, S. J., & Norvig, P. (1995). In *Artificial Intelligence: A Modern Approach* (1st ed.). Prentice Hall.

Saikat Dutt, S., & Geetha Chandramouli Subramanian, C. (2015). Software Engineering. https://learning.oreilly.com/library/view/software-engineering/9789332558298/xhtml/Chapter002.xhtml

Stanley, K. O. (2014, December 12). *NeuroEvolution of Augmenting Topologies*. The NeuroEvolution of Augmenting Topologies (NEAT) Users Page. http://www.cs.ucf.edu/~kstanley/neat.html

Stanley, K.O., D'Ambrosio, D., Gauci, J., 2009. A Hypercube-based indirect encoding for evolving large-scale neural networks. Artificial Life. Vol. 15:2. pp 185-212.

Stanley, K. O., & Miikkulainen, R. (2002a). *Efficient Reinforcement Learning through Evolving Neural Network Topologies*. 9.

Stanley, K. O., & Miikkulainen, R. (2002b). Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*, *10*(2), 99–127.

Stanley, K. O., & Miikkulainen, R. (2004). Competitive Coevolution through Evolutionary Complexification. *Journal of Artificial Intelligence Research*, *21*, 63–100.

Stanley, K., Kohl, N., Sherony, R., & Miikkulainen, R. (2005). Neuroevolution of an automobile crash warning system. *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*

Taud, H., & Mas, J. F. (2018). Multilayer Perceptron (MLP). In M. T. Camacho Olmedo, M. Paegelow, J.-F. Mas, & F. Escobar (Eds.), Geomatic Approaches for Modeling Land Change Scenarios (pp. 451–455). Springer International Publishing.

Tupper, A., & Neshatian, K. (2020). Evolving neural network agents to play atari games with compact state representations. *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, 99–100. https://doi.org/10.1145/3377929.3390072

van Rossum, G., & Coghlan, N. (2001, July 5). *PEP 8—Style Guide for Python Code*. Python.Org. https://www.python.org/dev/peps/pep-0008/

Vidya. (2018, November 5). 4 ways to visualize Likert Scales. *Daydreaming Numbers*. http://daydreamingnumbers.com/blog/4-ways-to-visualize-likert-scales/

Wahid, S., & Wahid, B. (2018). *The Effectiveness of Gamification in Improving Student Performance for Programming Lesson*.

Wexler, S. (2021, August 16). *How to visualize Likert scale data in Tableau*. Data Revelations. https://www.datarevelations.com/howto-likert/

Wittkamp, M., Barone, L., & Hingston, P. (2008). Using NEAT for continuous adaptation and teamwork formation in Pacman. *2008 IEEE Symposium On Computational Intelligence and Games*, 234–242.

wpjohn, user. (2017, June 9). COLORS. *WikiPython*. https://www.wikipython.com/tkinter-ttk-tix/summary-information/colors/

YAO, X., 1999. *Evolving artificial neural networks*

Yegulalp, S. (2021, October 27). *Virtualenv and venv: Python virtual environments explained. InfoWorld.* https://www.infoworld.com/article/3239675/virtualenv-and-venv-python-virtual-environments-explained.html

Örkcü, H. H., & Bal, H. (2011). Comparing performances of backpropagation and genetic algorithms in the data classification. *Expert Systems with Applications*, *38*(4), 3703–3709.

How to Present Likert Scale Data? An Ultimate Guide for Google Sheets Users. (n.d.). Retrieved November 23, 2021, from https://ppcexpo.com/blog/how-to-present-likert-scale-data

Welcome to NEAT-Python's documentation! —NEAT-Python 0.92 documentation. (n.d.). Retrieved November 23, 2021, from https://neat-python.readthedocs.io/en/latest/

8 Appendix 1 Project Overview

Venetsia Krasteva 40313507

Initial Project Overview

SOC10101 Honours Project (40 Credits)

Title of Project:

A graphical application to simplify the use of NEAT to evolve video game characters.

Overview of Project Content and Milestones

The aim of this project is to develop an easy-to-use graphical application that allow the user to evolve neural network characters for different Atari games.

Objectives:

1. Produce a literature review of research on the evolution of neural network characters for video games with a focus on the NEAT library.
2. Design the front-end (graphical user interface) for the application.
3. Design the software architecture i.e., how the application will connect with NEAT and the video game.
4. Implement the application.
5. Test the application on a variety of Atari video games.
6. Produce the dissertation document detailing the above work.

The Main Deliverable(s):

An easy-to-use application that evolves video game characters using the NEAT neural network library.

The Target Audience for the Deliverable(s):

Researchers that would like to evolve a neural network without any using any code. Members of the public interested in a demonstration in AI for video games.

The Work to be Undertaken:

- Investigate implementation of NEAT in evolving video game characters.
- Write a literature review.
- Specification of the requirements
- Produce the design for front-end and back-end of the application.
- Implementation of the design in a suitable programming language

SOC10101 Honours Project 05/02/2021 1

- Evaluate it with user studies to find out whether the application is easy to use.

Additional Information / Knowledge Required:

- New knowledge –Python Neat library, working with Neural networks, theoretical knowledge of neural networks and evolutionary algorithms.
- Extending current skills – building a software and writing a report
- Technologies - Stella (for Arcade Games)

Information Sources that Provide a Context for the Project:

Risi , S., & Togelius, J. (2015, November 11). *Neuroevolution in Games: State of the Art and Open Challenges*. https://arxiv.Org/. https://arxiv.org/pdf/1410.7326.pdf#page=18&zoom=100,416,102

M. Hausknecht, J. Lehman, R. Miikkulainen, and P. Stone. *A neuroevolution approach to general Atari game playing*. In IEEE Transactions on Computational Intelligence and AI in Games, 2013. https://ieeexplore-ieee-org.ezproxy.napier.ac.uk/stamp/stamp.jsp?tp=&arnumber=6756960

Omelianenko, I. (2019). *Hands-On Neuroevolution with Python*. Packt Publishing Ltd. https://learning.oreilly.com/library/view/hands-on-neuroevolution-with/9781838824914/cover.xhtml

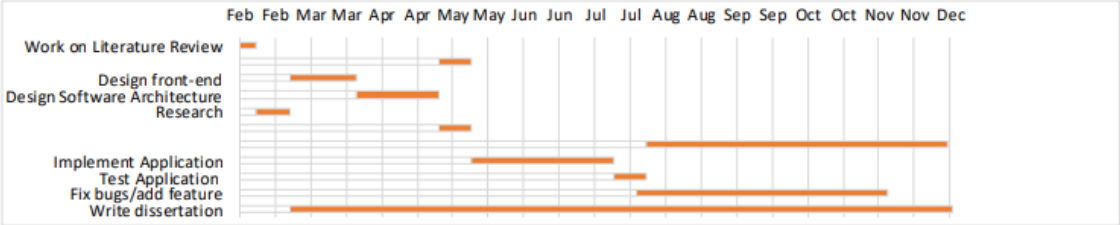
D. Levy, S. (2020, November 15). *NEAT-Gym*. Retrieved February 4, 2021, from https://github.com/simondlevy/neat-gym

The Importance of the Project:

The importance of the project is that it would be implemented in an easy-to-use design without writing any code in order to train a neural network to play the game. A similar project is already accessible in GitHub, but this that software you still need to write code to use it. This project aims to fill this gap and hence greatly increase the accessibility of NEAT to people who are not comfortable with coding and the command line.

The Key Challenge(s) to be Overcome:

Get all the components to work together.



9 Appendix 2 Second Formal Review Output

SOC10101 Honours Project (40 Credits)

Week 9 Report

Student Name: Venetsia Krasteva

Supervisor: Simon Powers

Second Marker: Andreas Steyven

Date of Meeting: 01/04/2021

Can the student provide evidence of attending supervision meetings by means of project diary sheets or other equivalent mechanism? **yes** **no***

If not, please comment on any reasons presented

Not seen these, but I take it that supervision meetings are happen regularly.

Please comment on the progress made so far

Progress so far is a bit behind where it should be.

A very interesting topic overall, but the focus is still a bit lacking. At this stage the aim and plan for evaluation should be fixed, so that the second half of the project can be used for implementation and evaluation.

Is the progress satisfactory? **yes** **no***

Can the student articulate their aims and objectives? **yes** **no***

If yes then please comment on them, otherwise write down your suggestions.

Although still a bit vague, the main aims and objectives were mentioned.

Building a system that lets someone without programming experience explore an machine learning algorithm. The system trains a neural network to control a games character in an Atari game.

* Please circle one answer; if **no** is circled then this **must** be amplified in the space provided

Does the student have a plan of work? **yes** **no***

If yes then please comment on that plan otherwise write down your suggestions.

There is an overall plan in form of a GANTT chart. However, due to a lack in focus it doesn't include the evaluation.

If, as discussed, the focus will be to develop an educational tool, then the evaluation might take a long time and needs to be carefully planned.

Does the student know how they are going to evaluate their work? **yes** **no***

If yes then please comment otherwise write down your suggestions.

Consider literature on gamification if your tool is educational. This might give you a higher response rate.

If you do a user study, make sure to get the appropriate consent forms signed from each participant. There are samples and guidance available. The university also has a survey server that you could use.

Make sure to cover the relevant theory in the methodology section for the evaluation method you use.

Defining your evaluation gives your project more focus as you develop with the aim to gather the necessary data.

Any other recommendations as to the future direction of the project

Make sure to include the chosen software development lifecycle and project management techniques and how you employ them to guide your project.

You don't need to review different techniques, but describe how they are the appropriate choice for your type of project.

* Please circle one answer; if **no** is circled then this **must** be amplified in the space provided

Signatures: Supervisor Second Marker AS

Student

The student should submit a copy of this form to Moodle immediately after the review meeting; A copy should also appear as an appendix in the final dissertation.

* Please circle one answer; if **no** is circled then this **must** be amplified in the space provided

10Appendix 3 Diary Sheets (or other project management evidence)

10.1 Meeting Notes (Sprints)

https://livenapierac-my.sharepoint.com/:f:/r/personal/40313507_live_napier_ac_uk/Documents/Dissertation%20Project%20Management/Meeting%20Notes?csf=1&web=1&e=6TpQxe - Full meeting notes can be found on Evernote file/HTML/PDF (same file)

15th of November 2021

- [ ] Use cases - written use case description
  - [ ] this is the actor and this is a graph of what they will do, we have a goal and preconditions, post-conditions
- [ ] Agile - include evidence of the meetings in (Evernote) Appendix (not that important as long as I highlight benefits of Agile)
- [x] ~~Testing - no unit test (tested on the way, after every task has been completed in an agile way) -> Validation for Testing~~
- [ ] Move project Plan (Chapter 3 currently) to the Appendix
- [x] ~~Implementation should be a start of a new chapter (Chapter 5)~~
  - [x] ~~combine implementation and testing~~
  - [ ] after object oriented and function oriented, should talk about changes within Neat and Atari-py (after whole design part)
- [ ] User evaluation (chapter 6)
  - [ ] present the survey first and explain the questions I have asked and why have I asked these certain questions and explain the use of the Liket Scale]
  - [ ] Methodology - Explain that the survey was hosted in NoviSurvey , explain how I recruited participants for the study
    - [ ] Statistics/ Graphs - etc. 90 % of the users agreed with tihs
- [ ] Overall Conclusion (chapter 7)

- [x] ~~Do not need to write too much about maintenance as it is not a commercial project~~
- [x] ~~testing should come after System Design~~
- [ ] Reference features of the application to the requirements
  - [x] ~~requirements should be in numerical order~~
- [ ] Include examples of education mode and talk about them (in implementation chapter)
  - [ ] Can saw for referencing: the lesson is drew on the material from

---

8th of November 2021

Questionnaire:

- [x] ~~Fix issues with questions~~

Connect form part 2:

- [x] ~~Remove for before may be submitted as it is not needed~~

- [x] ~~Distribute program to other people as all of it is ready~~

Dissertation needs to show off what I have done for the project, if something is not in the dissertation but I have indeed done it I will not get the marks for it (dissertation includes the evidence)

- [ ] Show different features in dissertation as markers would not run the code
  - [ ] include screenshots
- [ ] Explain the changes I have done in Neat and Atari-py library in Implementation Chapter (one after Design)
  - [ ] no need to screenshot code
    - [ ] **Screenshots of the application** running to highlight a feature that has been included
- [ ] Draw attention to the guidance in GitHub in Implementation Chapter, that I have produced it so people can clearly see how to install it and use app in the study

This project is more for education and research, requirements analysis is necessary but do not need pages and pages

---

1st of November 2021

Tasks to do:

- [x] ~~Fix questionnaire that I have designed~~
  - To what extend do you agree with this statement:
    - This application furthered my knowledge in Artificial Intelligence.
- [ ] Have a look at the Likert scale (quantitative data)

- [x] ~~Fix Neat Page in Education mode~~
- [x] ~~Remove comparison with backpropagation~~
- [x] ~~Specify that it is indeed Pixels for Atari games~~
- [ ] Fix Diagrams
- [x] ~~take buttons use case diagram out (or change name of the use case)~~
  - what does this page lets the user do
  - use case - what goal is the user trying to achieve
  - use case diagram **helps to model the system**, user interactions while activity diagram helps to model the work flow of the system.

Tasks done:

- included Neat Page in Education mode
- Fix Consent Form
- Requirements engineering and diagrams

In dissertation:

- [ ] Include finding for Recurrent vs Feed-Forward in Space Invaders
  - Why? - in space invaders there is no advantage to having time series data
- [ ] Use case diagrams are used to display a verb (something that the user wants to achieve)

22th of October 2021

- Demo the application

Questions:

- regarding the ethics

- Survey software (napier.ac.uk)
  - think about questions
- Consent form

Tasks done:

- tested all games in Atari library
- include guidance in GitHub
- make sticky notes short

Tasks to do:

- [x] ~~fix answers in questions for education mode~~

18th of October 2021

Things that are done:

- browse folder for the winner
- fixed some error that were in application
- irresponsible - implemented a thread pool so Neat executes on a different thread
- checkpoints are saved from 0 generation and every n generation

Things to do:

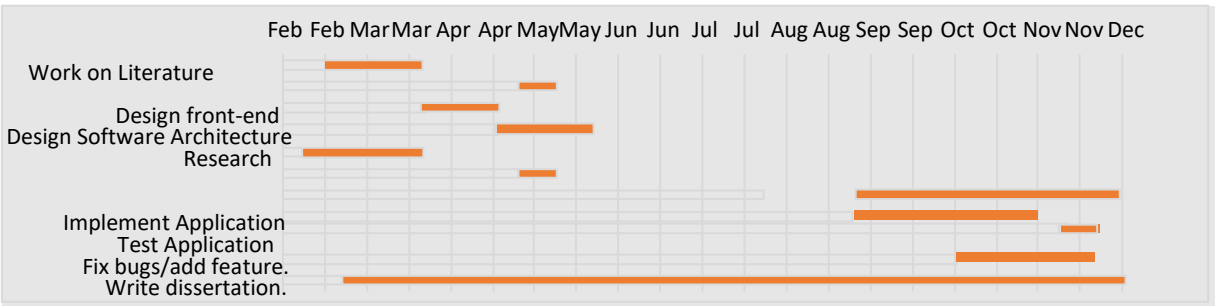
- [x] ~~fix checkpoints so user can see n number of genomes per checkpoint~~
- [x] ~~ethics for user evaluation~~
  - [x] ~~read all carefully~~
  - [x] ~~self assessment questionnaire~~
  - [x] ~~do survey via NoviServey~~
  - [x] ~~think about questions for survey~~
- [x] ~~try to get the executable working~~
- [x] ~~testing tables~~

10.2 Trello Board (Kanban) - https://trello.com/b/4EZnxsn7

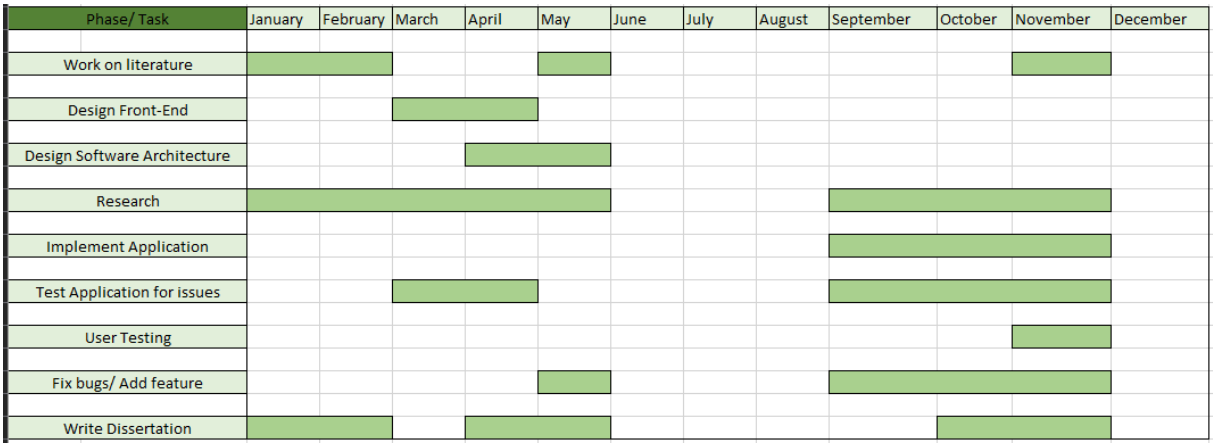
Figure 58 Trello (Kanban) QR Code (Snap of the board can be seen in Figure 21)

10.3 Plan

10.3.1 Old



10.3.2 New


Phase/ Task
January
February
March
April
May
June
July
August
September
October
November
December
Work on literature
Design Front-End
Design Software Architecture
Research
Implement Application
Test Application for issues
User Testing
Fix bugs/ Add feature
Write Dissertation

11Appendix 4 and following

11.1 Test Outputs for other tabs:

11.1.1 NEAT Config Test Outputs

Input	Expected	Passed	Issues	Resolution
Neat Config				
Gym Game				
Game_selection = SpaceInvaders-v0	Pass	YES		
Game_selection = Breakout-v0	Fail	No	The game is not working, so it should not be allowed to work	Create an "allow list"[25] of games
Game_selection = ""	Pass	Yes		
Neat Section - Mandatory				
Fitness_criterion = min	Pass	Yes		
Fitness_criterion = ""	Fail	No	Value must not be empty	Perform a check if value is empty
Fitness_criterion = average	Fail	No	Fitness_criterion accepted values are [max, min, mean]	Create an "allow list" of specific values
Fitness_threshold = 10	Pass	Yes		Value is checked if it is int or float
Fitness_threshold = ""	Fail	No	Value must not be empty	Perform a check if value is empty
Fitness_threshold = 10.5	Pass	No	Value can be float	Value is checked if it is int or float
Fitness_threshold = 10A	Fail	No	Fitness Threshold must be only integer/float (double) values	Check if string contains using any alphabetic

[25] Allow list - Allow list validation is appropriate for all input fields provided by the user. Allow list validation involves defining exactly what IS authorized, and by definition, everything else is not authorized.

				characters using regex if re.search('[a-zA-Z]', value)
Fitness_threshold = -10	Pass	No	Value can be negative	Perform an extra check if value contains character "-" on first character and check rest of string for int/float
Fitness_threshold = 1-0	Fail	No	Value cannot be of that type – must be float or int	After checks for minus character, check if value is int or float and fail if it isn't
Fitness_threshold = -10a	Fail	Yes		
Fitness_threshold = -10 a	Fail	Yes		
Fitness_threshold = !10	Fail	Yes		
Fitness_threshold = 5!	Fail	Yes		
Fitness_threshold = 0	Pass	No	Value can be 0	Perform a check if value is == to 0
Fitness_threshold = 0.0	Pass	No	Value can be 0.0	Perform a check if value is == to 0.0
Fitness_threshold = 0.1	Pass	Yes		
Fitness_threshold = 0.1A	Fail	Yes		
Fitness_threshold = 5 a	Fail	Yes		
No_fitness_termination, single_structural_	Pass	Yes		

mutation,enabled _default = False				
No_fitness_termination, single_structural_mutation,enabled _default = True	Pass	Yes		
No_fitness_termination, single_structural_mutation,enabled _default = Other	Fail	No	Value can be True or False only	Perform a check if value is true or false and set default value anything else
No_fitness_termination, single_structural_mutation,enabled _default = “”	Fail	No	Value must not be empty	Value is checked if it is empty and if yes, it sets the default value
No_fitness_termination, single_structural_mutation,enabled _default = 123	Fail	Yes		
Pop_size = 0	Fail	No	Value must not be 0	Check if value is bigger than 0
Pop_size = 1	Pass	Yes		
Pop_size = 0.5	Fail	No	Value must not be of type float – only integers (whole numbers) allowed	Check if value is of type int
Pop_size = a	Fail	Yes		
Pop_size = -1	Fail	No	Value must not be negative	Check if value contains the character “-“ and replace with empty the returned pop_size = 1
Pop_size = 1-12	Fail	Yes		
Pop_size = a1	Fail	Yes		
Pop_size = 1a	Fail	Yes		

Pop_size = 11 1	Fail	Yes		
Reset on extrinction = True	Pass	Yes		
Reset on extrinction = False	Pass	Yes		
Reset on extrinction = other	Fail	No	Value must be True or False	Check if value is True or False; otherwise, colour label red and delete value
Reset on extrinction = 1	Fail	Yes		
Reset on extrinction = “”	Fail	No	Value cannot be empty	Check if value is empty and if it is, delete value and colour label red
Default Stagnation				
Species fitness func = mean	Pass	Yes		
Species fitness func = max	Pass	Yes		
Species fitness func = min	Pass	Yes		
Species fitness func = median	Pass	Yes		
Species fitness func = other	Fail	No	Value must be mean, max, min or median	Perform a check if value is equal to one of [mean, max, min, median]
Species fitness func = 1	Fail	Yes		
Species fitness func = “”	Pass	No	Value cannot be empty	If value is empty, set the default one for the parameter

Max_stagination = 0	Pass	Yes		
Max_stagination = 15	Pass	Yes		
Max_stagination = a	Fail	No	Value must be an integer	Check if value is integer, and if it is not, the value is set to default value
Max_stagination = 0.5	Fail	Yes		
Max_stagination = -1	Fail	No	Value must not be negative	Check if value contains “-“ and replace with “” (empty)
Num_protected_species = a	Fail	No	Value must be integer	Check if value is integer
Num_protected_species = 0	Pass	No	Value can be zero	Make an exception for 0 as 0 is not an ingeteger
Num_protected_species = 2	Pass	Yes		
Num_protected_species = 0.2	Fail	Yes		
Num_protected_species = 1a	Fail	Yes		
Num_protected_species = -2	Pass	No	The value cannot be zero, but user can submit “-“ value as the “-“ is converted to “”	Convert “-“ to “”
Num_protected_species = -2a	Fail	Yes		
Default Reproduction				
Elitism = a	Fail	No	Value must be integer	Check if value is int
Elitism = 0	Pass	No	Value can be zero (default value is zero)	Try if value is int if not set value to 0 (default value)

Elitism = 0.0	Fail	Yes		
Elitism = 1a	Fail	Yes		
Elitism = 1.0	Fail	Yes		
Elitism = -1	Pass	No	Even if negative values are not accepted, the software will convert value to positive	Find if any “-“ and replace with “”
Elitism = 1 afd	Fail	Yes		
Survival_threshold = a	Fail	No	Value must be a float	Check if value is float; otherwise, set default value
Survival_threshold = 0	Pass	No	Value can be zero	Make an exception for 0 and 0.0
Survival_threshold = 0.0	Pass	Yes		
Survival_threshold = 12	Pass	No	Value must be of type float	Convert int to float
Survival_threshold = 12.0	Pass	Yes		
Survival_threshold = -12	Pass	No	Even though negative values are not allowed, program should convert to positive	Find “-“ in string and replace with “”
Survival_threshold = -12.0	Pass	Yes		
min_species_size = a	Fail	No	Value must be of type integer	Check if value is integer and if not, set the default value
min_species_size = 0	Fail	Yes		
min_species_size = 0.0	Fail	Yes		
min_species_size = 1.0	Fail	Yes		
Genome Section				
Num_input = a	Fail	No	Value must be integer	Check if value is integer
Num_input = 0	Fail	Yes		
Num_input = 0.0	Fail	Yes		

Num_input = - 15	Pass	No	Negative values are not allowed, but program will remove “-“	Find “-“ if any and replace with “”
Num_input = -15adsds	Fail	Yes		
Num_output = a	Fail	No	Value must be integer	Check if value is integer
Num_output = 0	Fail	Yes		
Num_output = 0.0	Fail	Yes		
Num_output = -15	Pass	No	Negative values are not allowed, but program will remove “-“	Find “-“ if any and replace with “”
Num_output = -15adsds	Fail	Yes		
Num_hidden = a	Fail	No	Value must be integer	Check if value is integer
Num_hidden = 0	Fail	Yes		
Num_hidden = 0.0	Fail	Yes		
Num_hidden = -15	Pass	No	Negative values are not allowed, but program will remove “-“	Find “-“ if any and replace with “”
Num_hidden = -15adsds	Fail	Yes		
Initial_connection = a	Fail	No	Value must be one of ["unconnected","fs_neat_nohidden”,"fs_neat_hidden",”full_nodirect","full_direct",”partial_direct", "partial_nodirect"]	Check if value is one of list; if not, choose default value
Initial_connection = unconnected	Pass	Yes		
Initial_connection = 1	Fail	Yes		
initial_connection_value,activation_mutate_rate,aggregation_mutate_rate,node_add_prob,node_delete_prob,enabled_mutate_rate,conn_add_prob,conn_delete_prob = 0	Pass	No	Value be zero but must be a float	Check if value is 0 or 0.0 and if yes, set to 0.0; otherwise, check if value is float

initial_connection_value,activation_mutate_rate,aggregation_mutate_rate,node_add_prob,node_delete_prob,enabled_mutate_rate,conn_add_prob,conn_delete_prob = 1	Pass	Yes		
initial_connection_value,activation_mutate_rate,aggregation_mutate_rate,node_add_prob,node_delete_prob,enabled_mutate_rate,conn_add_prob,conn_delete_prob = 1.0	Pass	Yes		
initial_connection_value,activation_mutate_rate,aggregation_mutate_rate,node_add_prob,node_delete_prob,enabled_mutate_rate,conn_add_prob,conn_delete_prob = a	Fail	Yes		
initial_connection_value,activation_mutate_rate,aggregation_mutate_rate,node_add_prob,node_delete_prob,enabled_mutate_rate,conn_add_prob,conn_delete_prob = 1.0a	Fail	Yes		
initial_connection_value,activation_mutate_rate,aggregation_mutate_rate,node_add_prob,node_delete_prob,enabled_mutate_rate,conn_add_prob,conn_delete_prob= - 1.0	Pass	No	Value cannot be harmful, but program will convert “-“ to “”	Find “-“ and replace to “”

feed_forward= True	Pass	Yes		
feed_forward= False	Pass	Yes		
feed_forward = other	Fail	No	Value must be True or False	Check if value is True or False; otherwise, colour label red and delete value
feed_forward = 1	Fail	Yes		
feed_forward = “”	Fail	No	Value cannot be empty	Check if value is empty and if it is, delete value and colour label red
Random_selector = True	Pass	Yes		
Random_selector = False	Pass	Yes		
Random_selector = other	Pass	Yes	Value must be True or False but does not cause any issues if not	Value must be True or False but does not cause any issues if not
Random_selector = 1	Pass	Yes		
Random_selector = “”	Pass	Yes		
Node activation and aggregation options				
activation_default = other	Fail	No	Value can be only one of: abs, clamped, cube, exp, hat, identity, inv, log, relu, elu, lelu, selu, sigmoid, sin, softplus, square or than	Check if value is one of the values in list, and if not, check if random_selector is enabled – if yes, choose randomly from list; if

				no, then leave empty
activation_default = abs	Pass	Yes		
activation_default = 1	Fail	Yes		
activation_default = “”	Fail	Yes		
aggregation_default= other	Fail	No	Value can be only one of: sum, product, min, max, mean, median, maxabs	Check if value is one of the values in list, and if not, check if random_selector is enabled – if yes, choose randomly from list; if no, then leave empty
aggregation_default= sum	Pass	Yes		
aggregation_default= 1	Fail	Yes		
aggregation_default= “”	Fail	Yes		
list_response_weight_bias[26] = a	Fail	No	Value must be of type float	Check if value is float by converting it
list_response_weight_bias = 12.0	Pass	Yes		
list_response_weight_bias = 12	Pass	Yes		
list_response_weight_bias = -12	Pass	No	Float does not allow “-“ so from the conversion, the -12 is converted into 12	Find if value has minus and set value as “- value.”

[26] list_response_weight_bias = enabled_rate_to_true_add,enabled_rate_to_false_add,enabled_rate_to_false_add,compatibility_weight_coefficient,compatibility_threshold,compatibility_disjoint_coefficient,bias_init_mean,bias_init_stdev,bias_max_value,bias_min_value,bias_mutate_power,bias_mutate_rate,bias_replace_rate,response_init_stdev,response_max_value, response_min_value,response_mutate_power,response_mutate_rate,response_replace_rate, response_init_mean, weight_init_mean,weight_init_stdev,weight_max_value,weight_min_value,weight_mutate_power, weight_mutate_rate,weight_replace_rate

list_response_weight_bias = -12.0	Pass	Yes		
list_response_weight_bias = -12.0a	Fail	Yes		
"bias_init_type","weight_init_typ","response_init_type" = a	Fail	No	Value must be one of 'gaussian', 'normal', 'uniform'	Check string if it is one of list if not set to default
“bias_init_type","weight_init_typ","response_init_type" = 1	Fail	Yes		
“bias_init_type","weight_init_typ","response_init_type" = 'gaussian'	Pass	Yes		
structural_mutation_surer = a	Fail	No	Value must be one of: default, False, True	Perform a check if value is one of the allowed ones if not set to default
structural_mutation_surer = True	Pass	Yes		
structural_mutation_surer = 1	Fail	Yes		

Table 17 Tests Output for Neat Config

11.1.2 Load Winner Test Outputs:

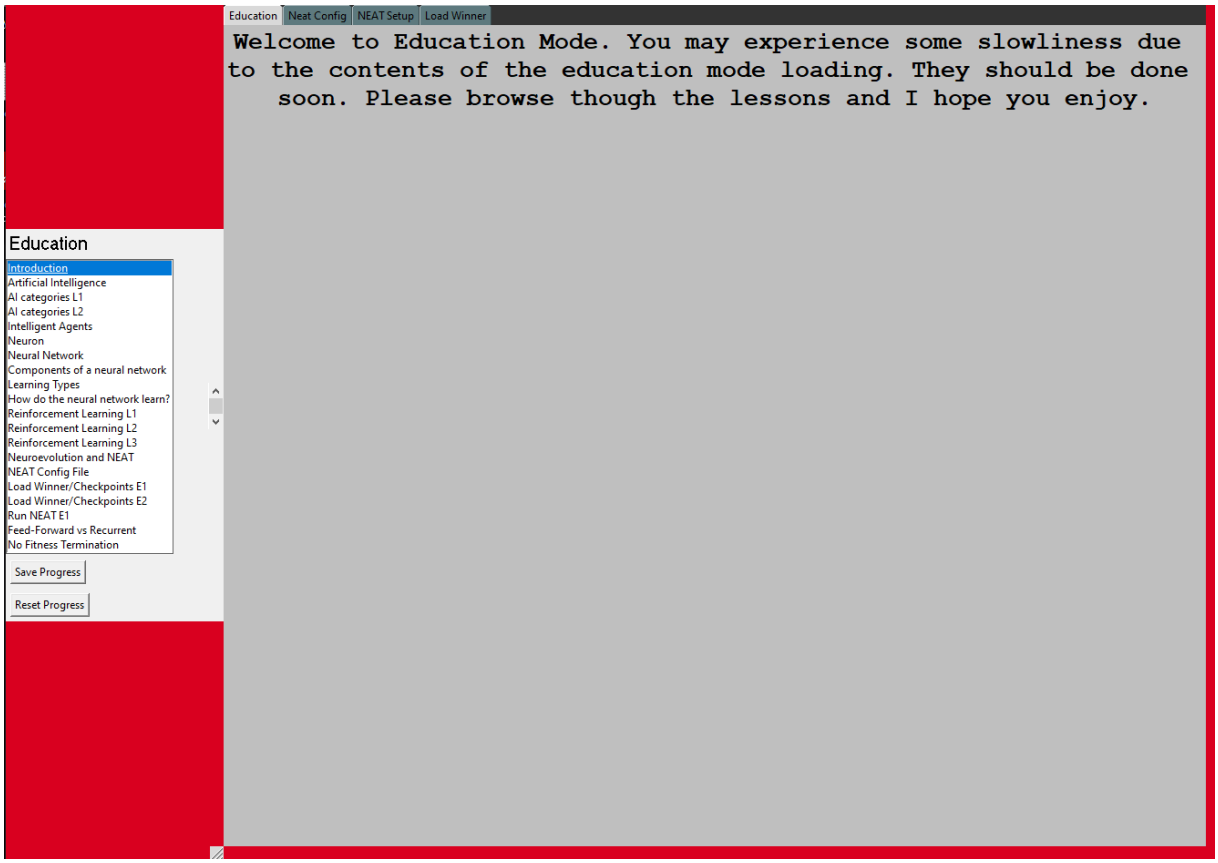
Input	Expected	Passed	Issues	Resolution
Game_selection = “”	Fail	No	Value cannot be empty	Define a list of accepted games and see if value is one of the values in list
Game_selection = a	Fail	Yes		
Game_selection = 1	Fail	Yes		
Game_selection = SpaceInvaders-v0	Pass	Yes		
Game_selection = Breakout-v0	Fail	Yes		
Winnerfilename = “”	Pass	Yes		
Winnerfilename = a	Pass	Yes		

Winnerfilename = a!	Fail	No	Special characters are not allowed in order not to make issues when getting path to winner file	On entering a character in field, the textbox checks the input and deletes any special characters
Winnerfilename = a1	Pass	Yes		
Winnerfilename = a 1	Fail	No	There should be no spaces	On entering a character in field, the textbox checks the input and deletes spaces
number_of_genomes= 0	Fail	No	Value cannot be zero	Perform a check if value is int and if not set to 1
number_of_genomes = a	Fail	Yes		
number_of_genomes = 1	Pass	Yes		
number_of_genomes = 1.0	Fail	No	Value cannot be of type float	Check if value is valid float and convert to int
number_of_genomes = 1.0a	Fail	Yes		
number_of_genomes = 0.0	Fail	No	Value is not valid float	Check if value is 0 or 0.0 and set to 1
number_of_genomes = -1	Fail	No	No negative values	Replace “-“ with “”
Network_type = “”	Fail	No	Value should not be empty	Color label red
Network_type = a	Fail	No	Value should be of a certain list	define an allowed list and check if value is one of them
Network_type = Feed-forward	Pass	Yes		
Network_type = Reccurrent	Pass	Yes		
Network_type = 1	Fail	Yes		
directory_value = <any manual input>	Fail	No	User should not be able to modify path	bind('<Key>',lambda e: 'break') – Bind widget to any input and prevent

directory_value = <chosen path from prompt>	Pass	No	Widget prevents any inputs	Insert text into widget by enabling the widget and then locking it
ep_per_genome = 0	Fail	No	Value cannot be zero	Perform a check if value is int and if not set to 1
ep_per_genome = a	Fail	Yes		
ep_per_genome = 1	Pass	Yes		
ep_per_genome = 1.0	Fail	No	Value cannot be of type float	Check if value is valid float and convert to int
ep_per_genome = 1.0a	Fail	Yes		
ep_per_genome = 0.0	Fail	No	Value is not valid float	Check if value is 0 or 0.0 and set to 1
ep_per_genome = -1	Fail	No	No negative values	Replace “-“with “”

Table 18 Tests Outputs Load Winner

11.2 Education Mode Lessons:



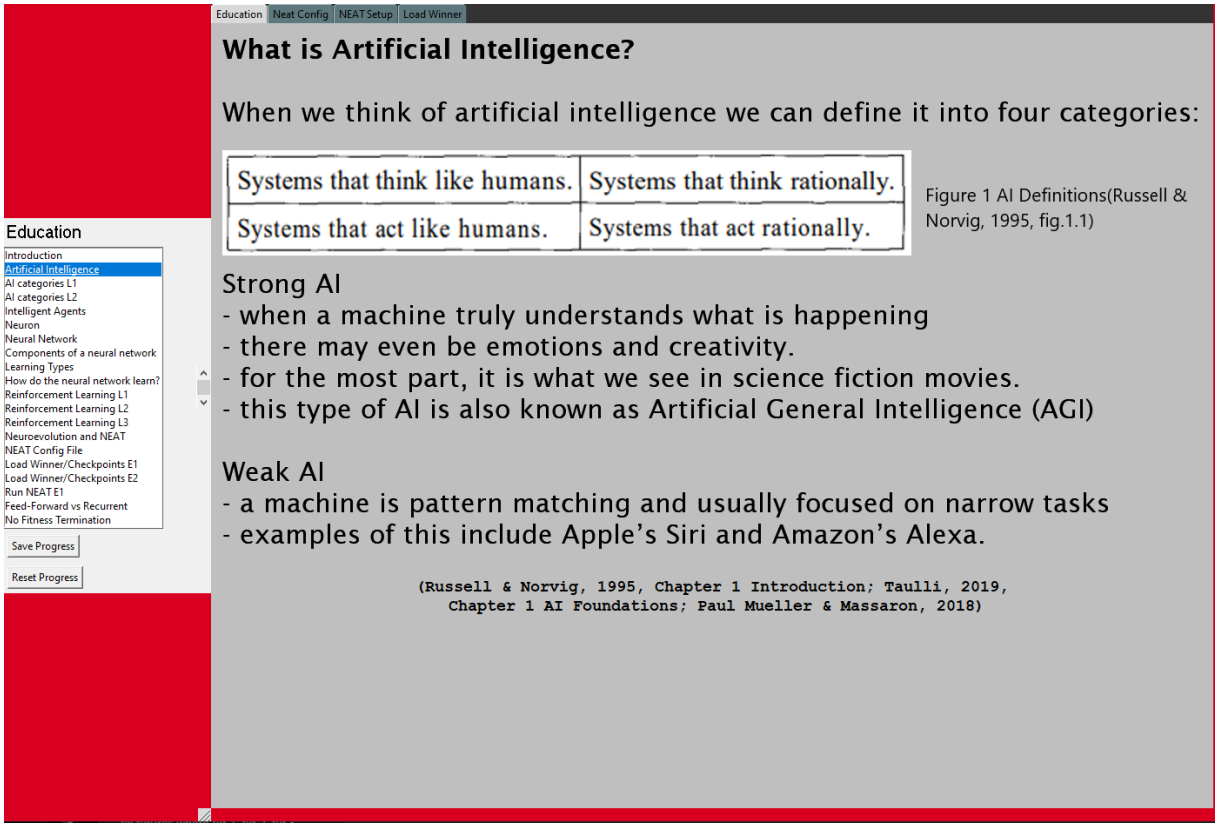
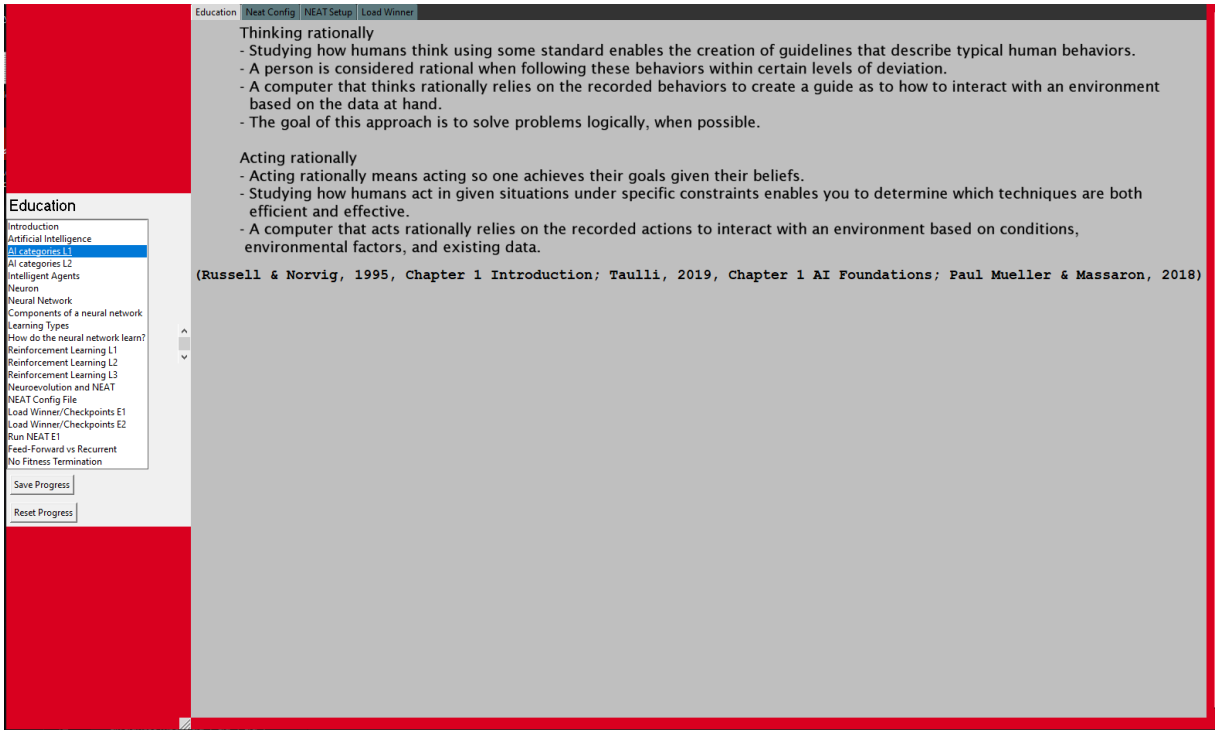
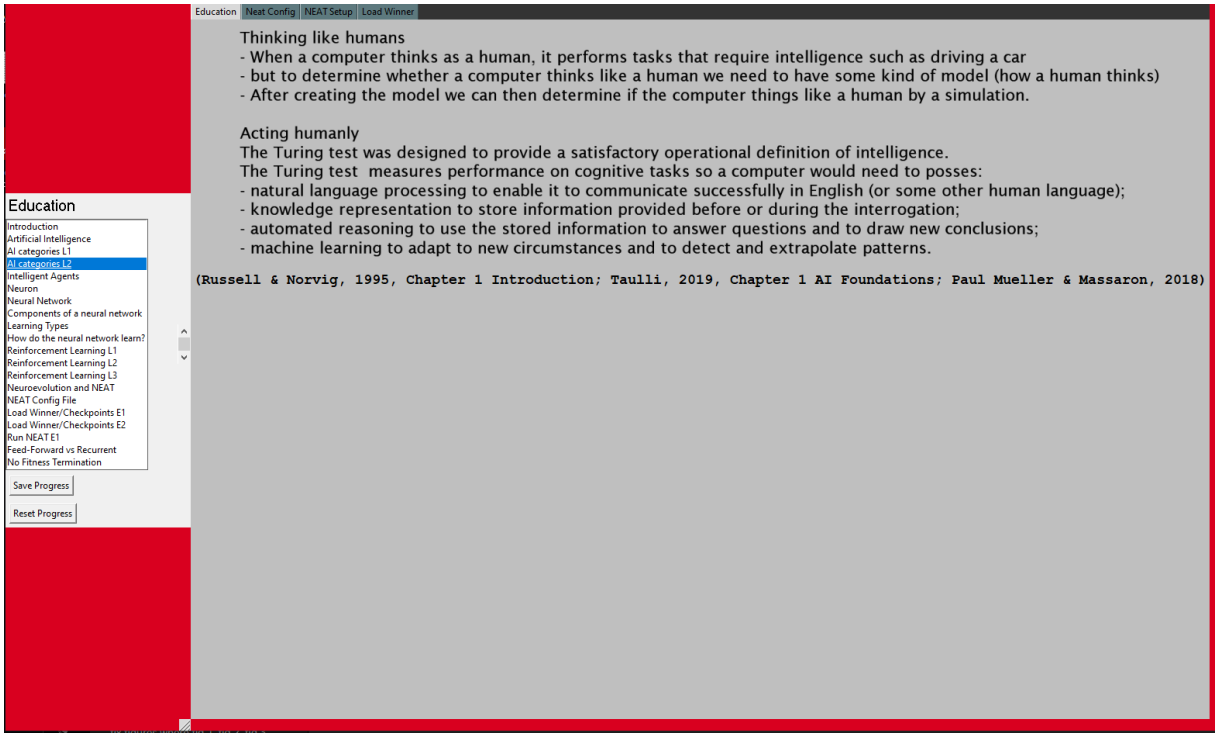
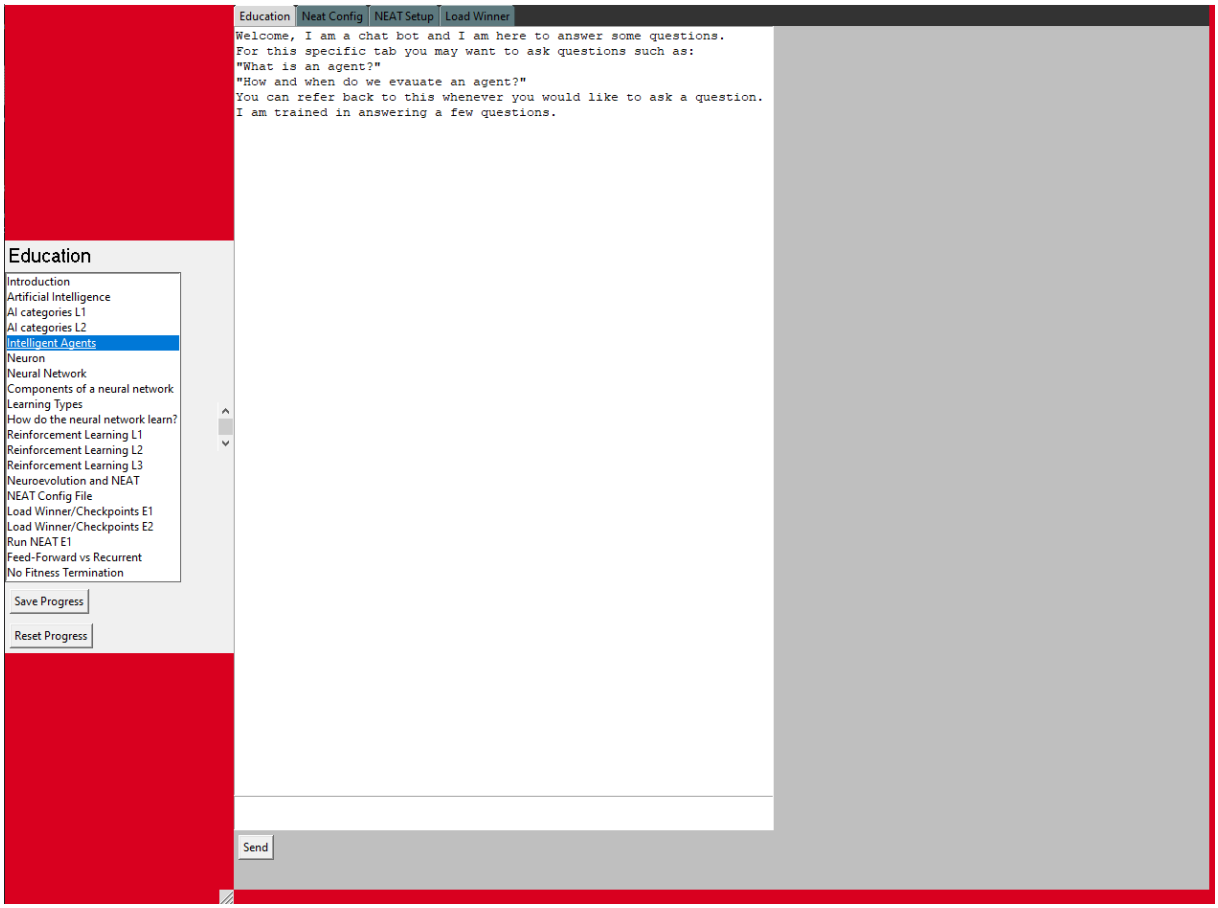


Figure 1 AI Definitions(Russell & Norvig, 1995, fig.1.1)




Education
Neat Config
NEAT Setup
Load Winner
Education
Introduction
Artificial Intelligence
AI categories L1
AI categories L2
Intelligent Agents
Neuron
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Save Progress
Reset Progress
Thinking like humans
- When a computer thinks as a human, it performs tasks that require intelligence such as driving a car
- but to determine whether a computer thinks like a human we need to have some kind of model (how a human thinks)
- After creating the model we can then determine if the computer things like a human by a simulation.
Acting humanly
The Turing test was designed to provide a satisfactory operational definition of intelligence.
The Turing test measures performance on cognitive tasks so a computer would need to posses:
- natural language processing to enable it to communicate successfully in English (or some other human language);
- knowledge representation to store information provided before or during the interrogation;
- automated reasoning to use the stored information to answer questions and to draw new conclusions;
- machine learning to adapt to new circumstances and to detect and extrapolate patterns.
(Russell & Norvig, 1995, Chapter 1 Introduction; Taulli, 2019, Chapter 1 AI Foundations; Paul Mueller & Massaron, 2018)


Education
Neat Config
NEAT Setup
Load Winner
Welcome, I am a chat bot and I am here to answer some questions.
For this specific tab you may want to ask questions such as:
"What is an agent?"
"How and when do we evauate an agent?"
You can refer back to this whenever you would like to ask a question.
I am trained in answering a few questions.
Send
Education
Introduction
Artificial Intelligence
AI categories L1
AI categories L2
Intelligent Agents
Neuron
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Save Progress
Reset Progress

Education | Neat Config | NEAT Setup | Load Winner

Education

- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

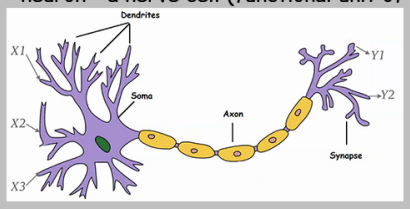
Save Progress

Reset Progress

Before we look into artificial neural network. We have to understand how were they created.

How was the artificial neural network created?

- inspired by the biological neuron found in the brain of animals
- neuron - a nerve cell (functional unit of all nervous system tissue)



Biological neurons receive short electrical impulses called signals from other neurons via these synapses. When a neuron receives sufficient signals from other neurons within a few milliseconds, it fires its signals.

Notice on the picture above we can see that there are some X's (inputs) and Y's (outputs).

(Russell & Norvig, 1995, Chapter 19.1 How the brain works)

Education | Neat Config | NEAT Setup | Load Winner

Education

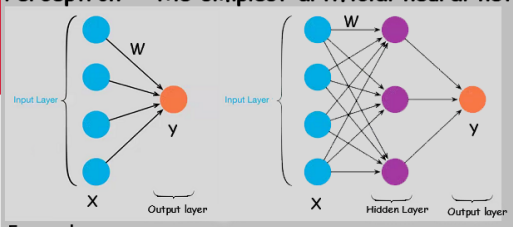
- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

Save Progress

Reset Progress

What is a artificial neural network?

Perceptron - the simplest artificial neural network



Example:
Consider we are trying to teach our nerual network to classify a cat via images.
Each input will be a feature of the cat. For example: its pointy ears, etc..
The weight ("W") would be initiallised depending on how much is the feature essential.
For example for input ears we would say above 0.8. But if say our input is "it's gray",
then this will not help the neural net learn better so we can initialise its weight to 0.01.

Question: What is the main difference between the Perceptron and the Multi-Layered Network?

Check answer

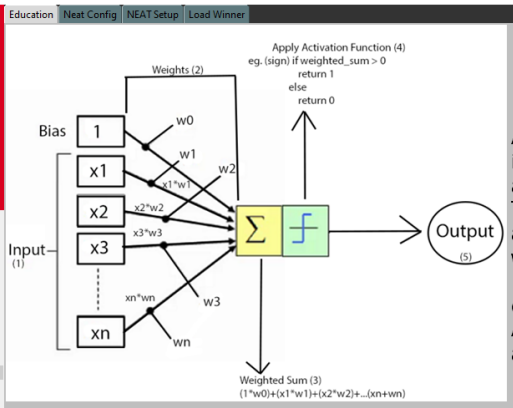
Education | Neat Config | NEAT Setup | Load Winner

Education

- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

Save Progress

Reset Progress



A neural network consists of units (nodes), inputs and output.The (1) input and the output are connected by links that have a numeric weight The (2) weight influences the strength of input and multiplying input and weight results in a (3) weighted input (sum). A (4) activation function is applied to the weighted sum, which forms the (5) output.
All these components form a perceptron - a network.

Simplest ANN, Source: (Adapted from Russell & Norvig, 1995, Fig 19.4; Daume III, 2012, Chapter the Perceptron Fig. 3.2)

Education | Neat Config | NEAT Setup | Load Winner

Education

- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

Save Progress

Reset Progress

Category	Supervised		Unsupervised		Reinforcement
Input and Output	Feed network with labelled input and output (encoded as pairs)		Feed network with labelled input with no output data		Agent interacting with the environment [do not have access to correct outputs that are connected to the input]
Usage	Classification	Regression	Clustering	Anomaly Detection	Games, Robots
AIM	Predict categories of examples based on features (class)	Predict continuous values	Grouping similar objects together	Identification of rare items or events	Train an agent in an environment based on rewards and penalties
Overview	Learning from examples		Discovering underlying patterns that we may not see		Learning behaviour by interacting in an environment and receiving rewards (positive or negative)

Source: (Adapted from Russell & Norvig, 1995, Chapter 18; Russell & Norvig, 1995, Chapter 20; Brown, 2014, pt. Neural Networks for Prediction)

Q: Can you guess what learning algorithm is NEAT?

Check answer

Education | Neat Config | NEAT Setup | Load Winner

Education

- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

Save Progress

Reset Progress

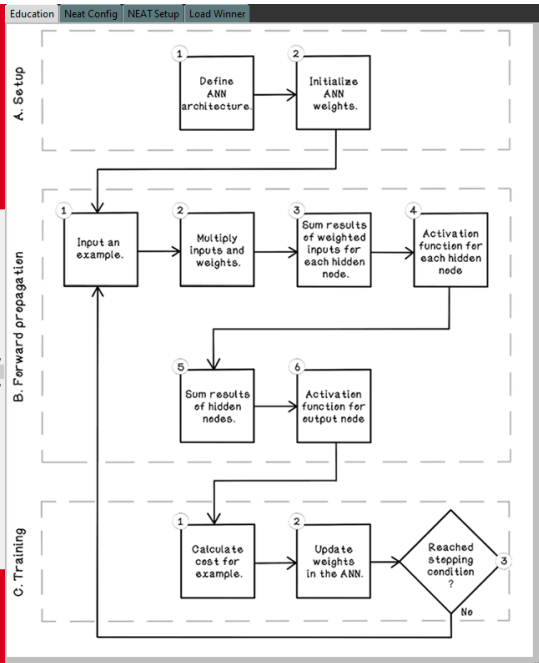


Figure 6 Life Cycle of training an ANN (Hurbans, 2020, fig. 9.20)

Phase: Setup
In this phase we start with defining the input nodes and the output nodes,the number of hidden layers and the number of neurons in each hidden layer, the activation functions user and mode. The weights in the ANN must be initialized to some value and then they will be constantly adjusted as the ANN learns. We will see this when we are writing the config file for NEAT.

Phase: Forward Propagation
Here we will input an example and perform calculations. In general the predicted output will be compared with a class of each example in the training set to train the network.

Phase: Training
Calculating the cost for the example is the difference between the predicted output and the actual class for the examples in the training set. The cost determines how bad the ANN is at predicting the class of examples.The weights will then be adjusted accordingly to be larger or smaller affecting the strength of the input. Because training cannot be infinite we have to determine when to stop. We initialise the training to stop after 1,000 iterations.

(Hurbans, 2020, Chapter 9 Artificial Neural Networks)

Education | Neat Config | NEAT Setup | Load Winner

Education

- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

Save Progress

Reset Progress

As you may already know NEAT is a reinforcement learning algorithm so fromhere we will focus mainly on this aspect of Artificial Intelligence. Up until now we have covered some basic things regarding Artificial Intelligence.

Now we will see how an agent learns as the here there is a goal but no examples where as with classical methods the network learns from examples.

What is Reinforcement Learning?
It is a task (no human labels data and no human collects or explicitly designs the collection of data) of learning though trial and error. The goal of the network trained is to act. It is concerned with long-term consequences of actions in anenvironment so the sequence of actions are important.

(Hubans, 2020, Chapter 10 Reinforcement Learning With Q-Learning;
Morales, 2020, Chapter 2 Mathematical Foundations of reinforcement learning)

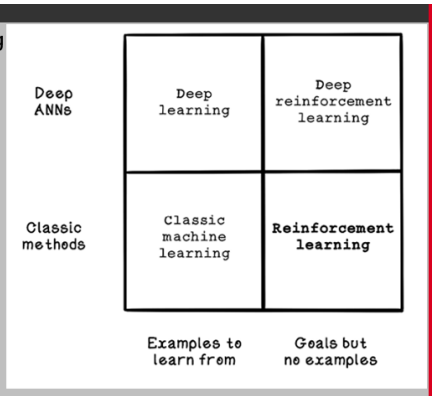


Figure 7 Categorization in learning types (Hurbans, 2020, fig. 10.2)

Education | Neat Config | NEAT Setup | Load Winner

Education

- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

Save Progress

Reset Progress

Where does it come from?

It comes from the behavioral psychology that explains behavior by a reflex action or something learned in the individual's history. Learning is based on rewards (positive reinforcement) or punishments(negative reinforcement), motivators for behaviors and aspects of the individual's environment.

(Hubans, 2020, Chapter 10 Reinforcement Learning With Q-Learning; Morales, 2020, Chapter 2 Mathematical Foundations of reinforcement learning)

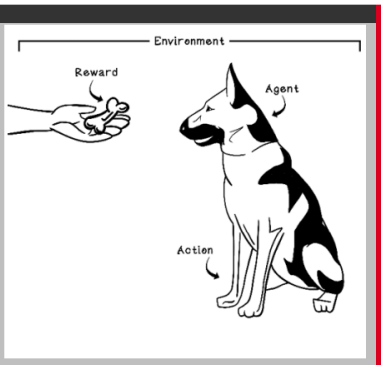


Figure 8 Example of reinforcement learning (Hurbans, 2020, fig. 10.3)

Education | Neat Config | NEAT Setup | Load Winner

Education

- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

Save Progress

Reset Progress

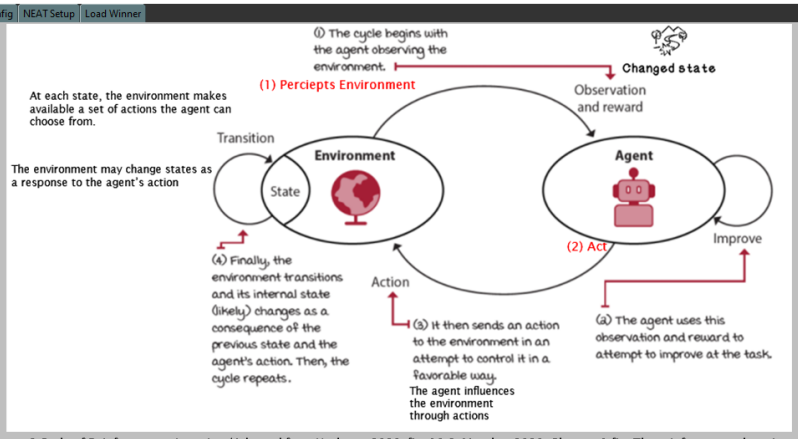


Figure 9 Cycle of Reinforcement Learning (Adapted from Hurbans, 2020, fig. 10.6; Morales, 2020, Chapter 1 fig. The reinforcement learning cycle)

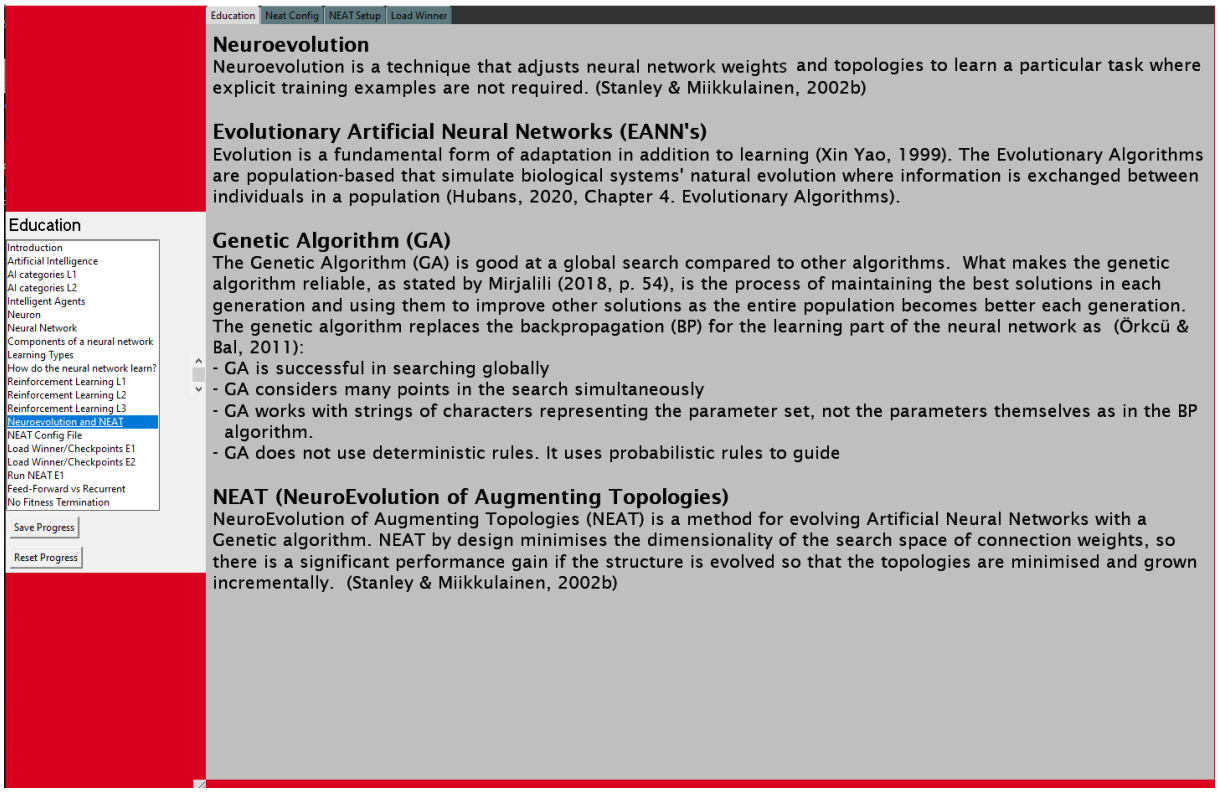
Like previously seen, a reinforcement learning model needs to be trained before we can use it. How will we do that with no example data?

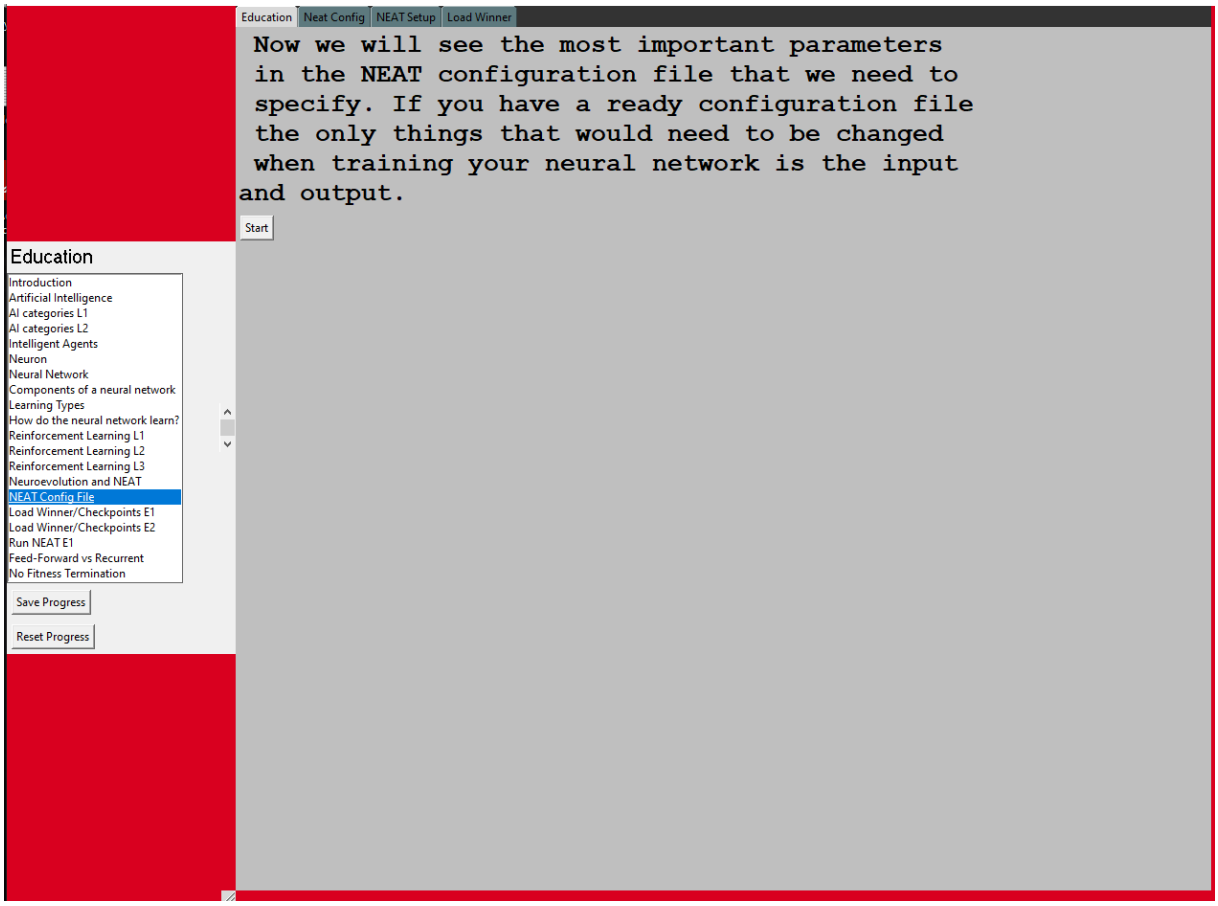
The training phase centers on exploring the environment and receiving feedback, given specific actions performed in specific circumstances or states.

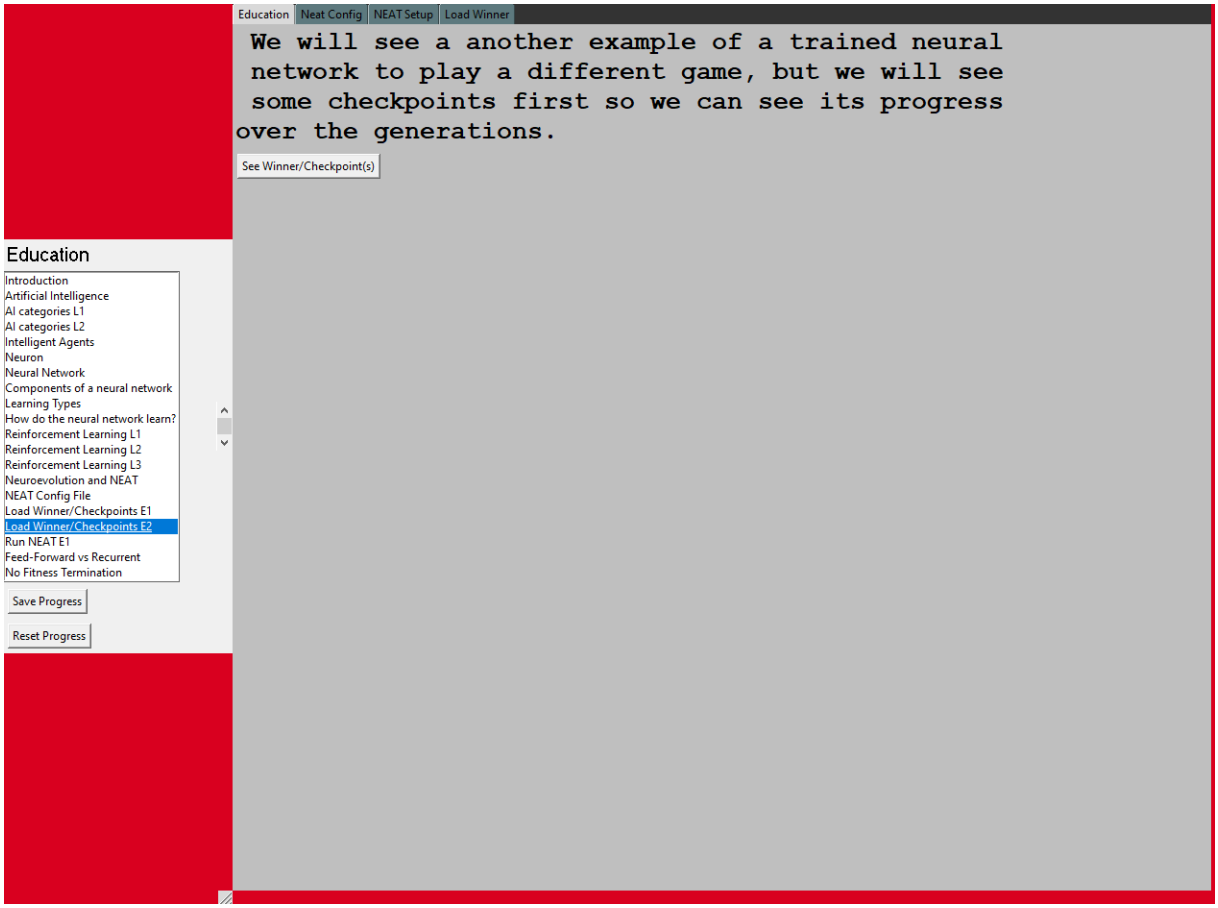
1. Initialise the environment - resetting the environment including the agent to the starting state.
2. Get the current state of the environment - function that provides current state of the environment that will change after each action is performed.
3. Apply an action to the environment - the agent applies some action to the environment and however the environment is affected by the action may result in reward, penalty or nothing.
4. Calculate the reward of the action - the reward/penalty/nothing has to be calculated for that action and effect on the environment.
5. Determine whether the goal is achieved - this function will determine if the agent is at its goal state.

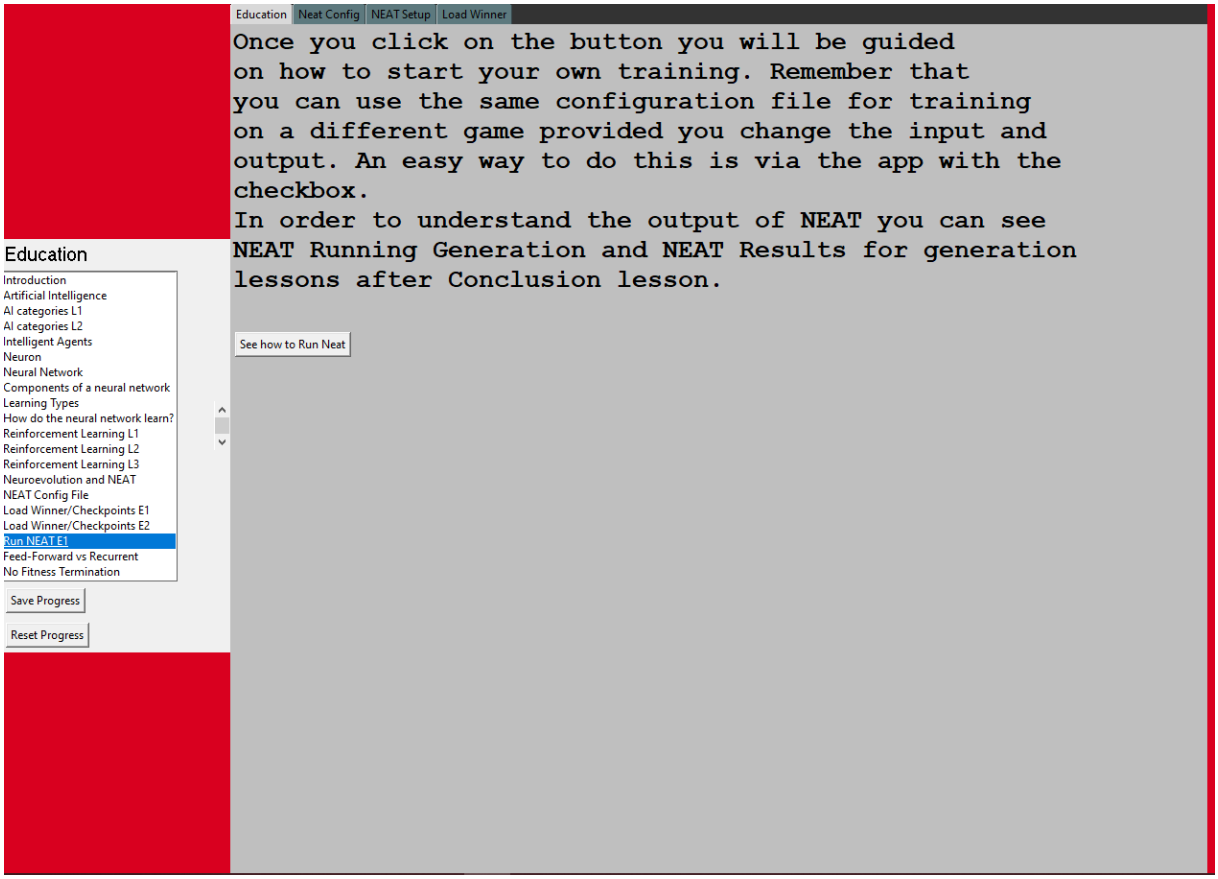
Once the environment is initialised it will go though the other 4 steps and repeat them until it either dies, runs out of time, reaches a goal state, etc. It will not initialise the environment after each action because (in most cases) it cannot reach goal state from only one action.

(Hubans, 2020, Chapter 10 Reinforcement Learning With Q-Learning; Morales, 2020, Chapter 2 Mathematical Foundations of reinforcement learning)


Education
Neat Config
NEAT Setup
Load Winner
Education
Introduction
Artificial Intelligence
AI categories L1
AI categories L2
Intelligent Agents
Neuron
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Save Progress
Reset Progress
Neuroevolution
Neuroevolution is a technique that adjusts neural network weights and topologies to learn a particular task where explicit training examples are not required. (Stanley & Miikkulainen, 2002b)
Evolutionary Artificial Neural Networks (EANN's)
Evolution is a fundamental form of adaptation in addition to learning (Xin Yao, 1999). The Evolutionary Algorithms are population-based that simulate biological systems' natural evolution where information is exchanged between individuals in a population (Hubans, 2020, Chapter 4. Evolutionary Algorithms).
Genetic Algorithm (GA)
The Genetic Algorithm (GA) is good at a global search compared to other algorithms. What makes the genetic algorithm reliable, as stated by Mirjalili (2018, p. 54), is the process of maintaining the best solutions in each generation and using them to improve other solutions as the entire population becomes better each generation. The genetic algorithm replaces the backpropagation (BP) for the learning part of the neural network as (Örkcü & Bal, 2011):
- GA is successful in searching globally
- GA considers many points in the search simultaneously
- GA works with strings of characters representing the parameter set, not the parameters themselves as in the BP algorithm.
- GA does not use deterministic rules. It uses probabilistic rules to guide
NEAT (NeuroEvolution of Augmenting Topologies)
NeuroEvolution of Augmenting Topologies (NEAT) is a method for evolving Artificial Neural Networks with a Genetic algorithm. NEAT by design minimises the dimensionality of the search space of connection weights, so there is a significant performance gain if the structure is evolved so that the topologies are minimised and grown incrementally. (Stanley & Miikkulainen, 2002b)


Education
Neat Config
NEAT Setup
Load Winner
Now we will see the most important parameters in the NEAT configuration file that we need to specify. If you have a ready configuration file the only things that would need to be changed when training your neural network is the input and output.
Start
Education
Introduction
Artificial Intelligence
AI categories L1
AI categories L2
Intelligent Agents
Neuron
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Save Progress
Reset Progress


Education
Neat Config
NEAT Setup
Load Winner
We will see a another example of a trained neural network to play a different game, but we will see some checkpoints first so we can see its progress over the generations.
See Winner/Checkpoint(s)
Education
Introduction
Artificial Intelligence
AI categories L1
AI categories L2
Intelligent Agents
Neuron
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Save Progress
Reset Progress


Education
Neat Config
NEAT Setup
Load Winner
Once you click on the button you will be guided on how to start your own training. Remember that you can use the same configuration file for training on a different game provided you change the input and output. An easy way to do this is via the app with the checkbox.
In order to understand the output of NEAT you can see NEAT Running Generation and NEAT Results for generation lessons after Conclusion lesson.
See how to Run Neat
Education
Introduction
Artificial Intelligence
AI categories L1
AI categories L2
Intelligent Agents
Neuron
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Save Progress
Reset Progress

Education | Neat Config | NEAT Setup | Load Winner

Education

- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

Save Progress

Reset Progress

Feed-Forward network	**Recurrent network**
links are unidirectional	the links can form arbitrary topologies.
no cycles	cycles back
information within them moves only in one direction (forward) (from the input nodes (units) towards the output units.)	using information from a previous forward pass over the neural network (share parameters across each layer of the network)
computes a function of the input values that depends on the weight settings (it has no internal state other than the weights themselves)	recurrent networks have internal state stored in the activation levels of the units
inputs and outputs are independent of each other	take information from prior inputs to influence the current input and output (the output depends on the prior elements within the sequence.)

(Adapted from *What Are Recurrent Neural Networks?*, 2021; Russell & Norvig, 1995)

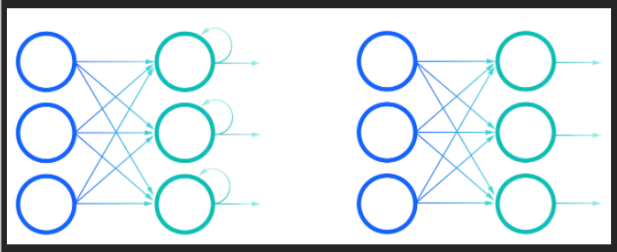


Figure 8 Recurrent (left) vs Feed-forward neural network(right) (*What Are Recurrent Neural Networks?*, 2021, fig. 1)

Education | Neat Config | NEAT Setup | Load Winner

Using no_fitness_termination you can specify for how many generations to run the algorithm for. The parameter has to be no_fitness_termination = True in config file.
This is a clever trick to use in order to be able to stop the algorithm and not leave it running without any progress. Currently there is no way to stop the algorithm from running within the app so application has to be closed completely if you want to start a different training

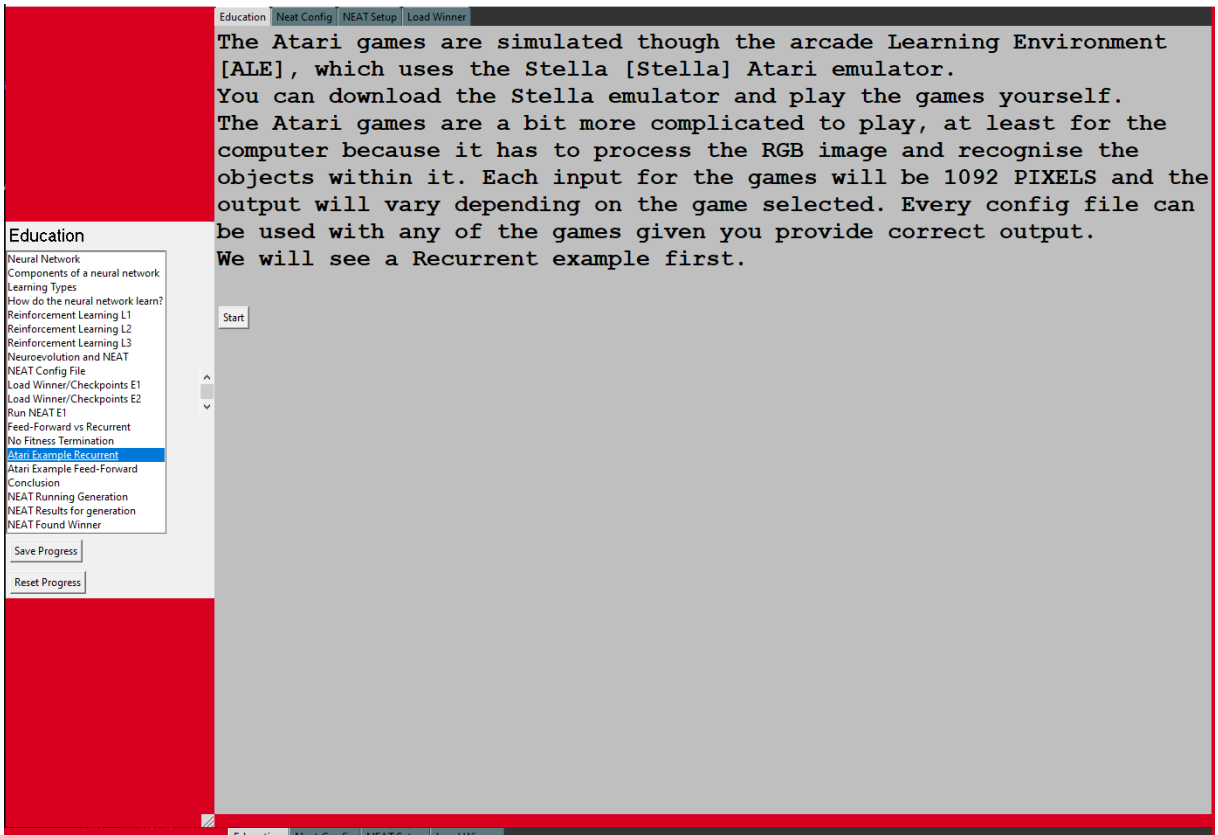
Start

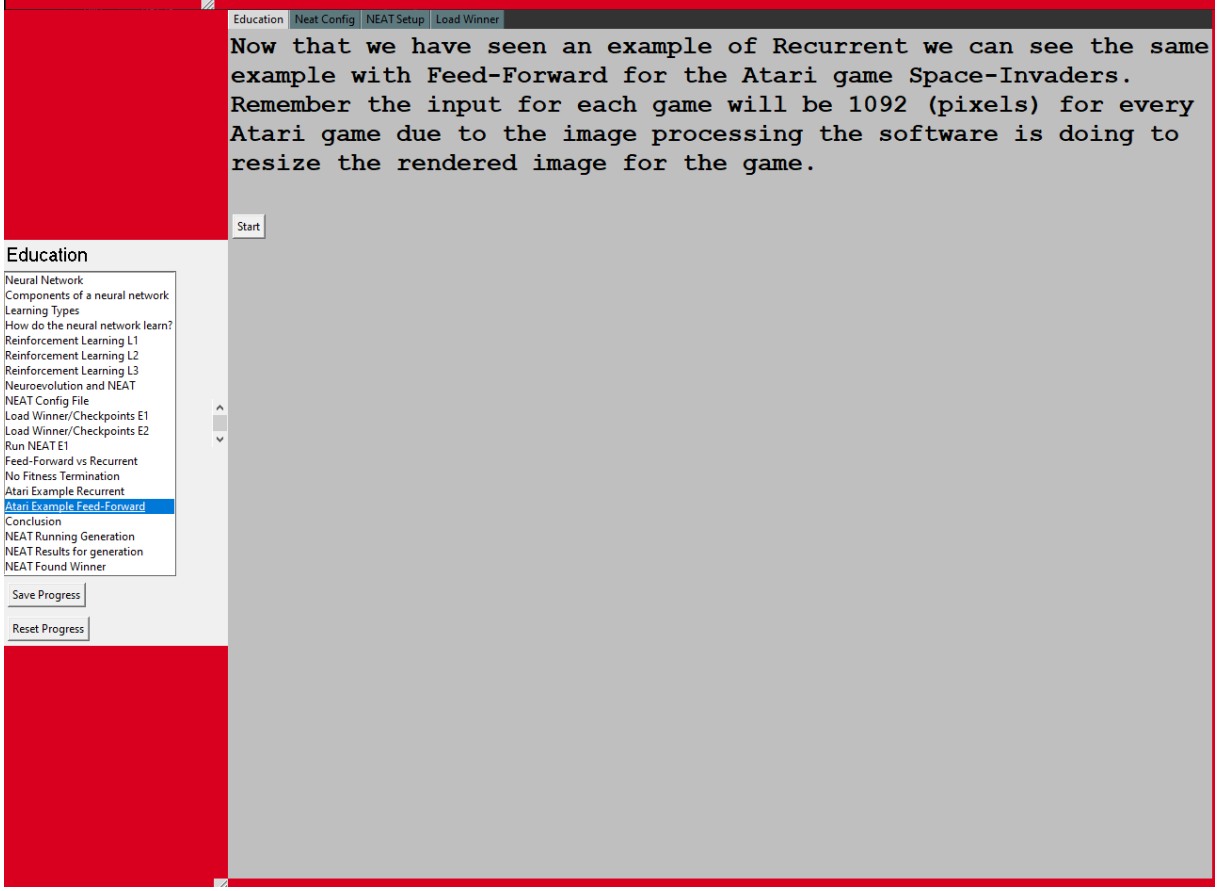
Education

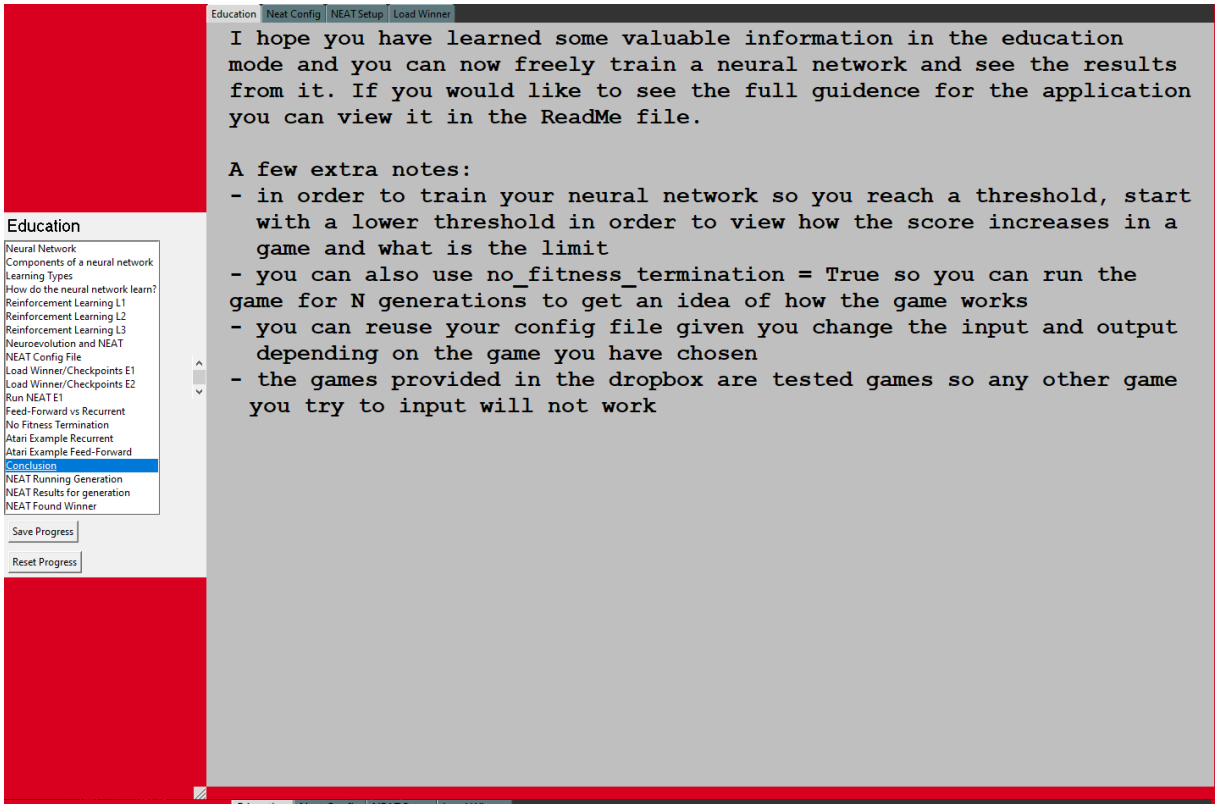
- Introduction
- Artificial Intelligence
- AI categories L1
- AI categories L2
- Intelligent Agents
- Neuron
- Neural Network
- Components of a neural network
- Learning Types
- How do the neural network learn?
- Reinforcement Learning L1
- Reinforcement Learning L2
- Reinforcement Learning L3
- Neuroevolution and NEAT
- NEAT Config File
- Load Winner/Checkpoints E1
- Load Winner/Checkpoints E2
- Run NEAT E1
- Feed-Forward vs Recurrent
- No Fitness Termination

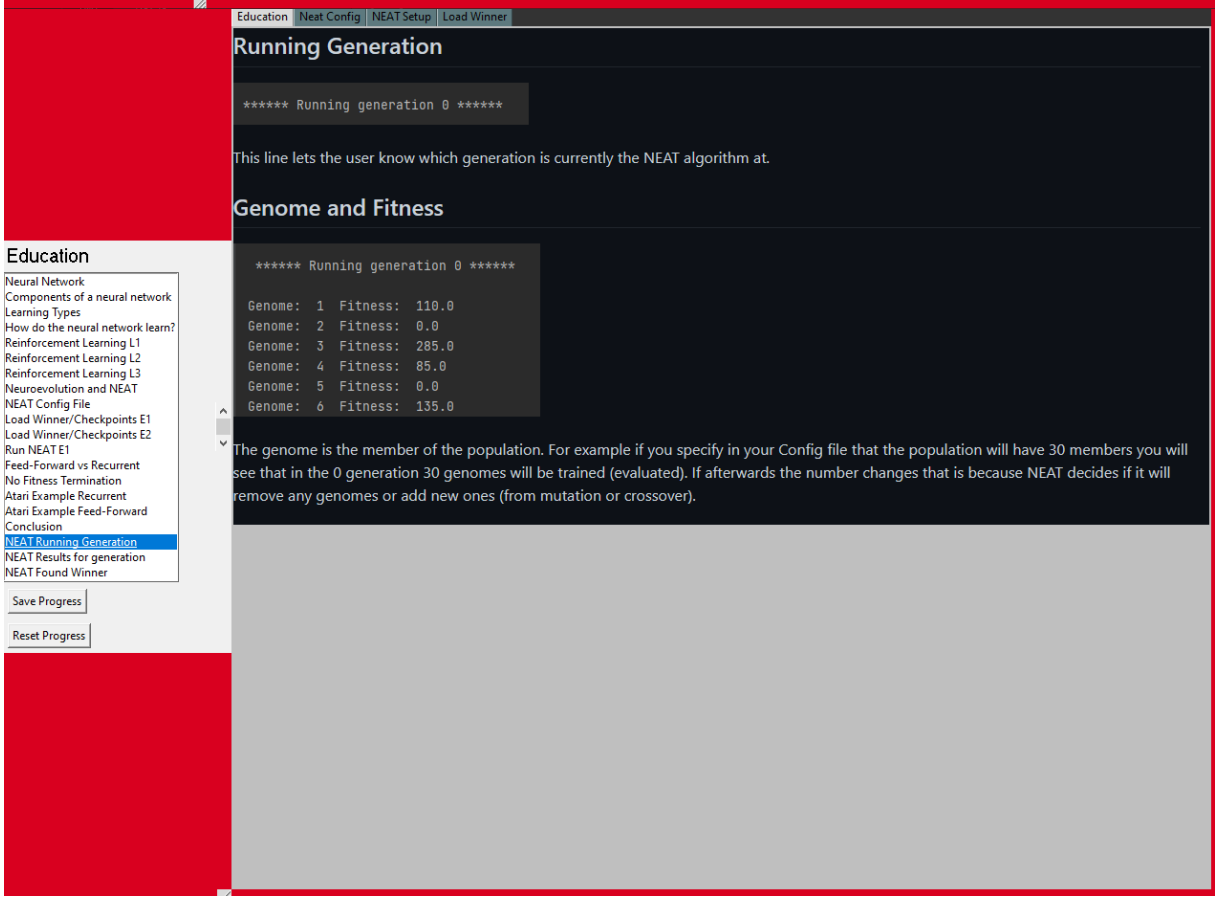
Save Progress

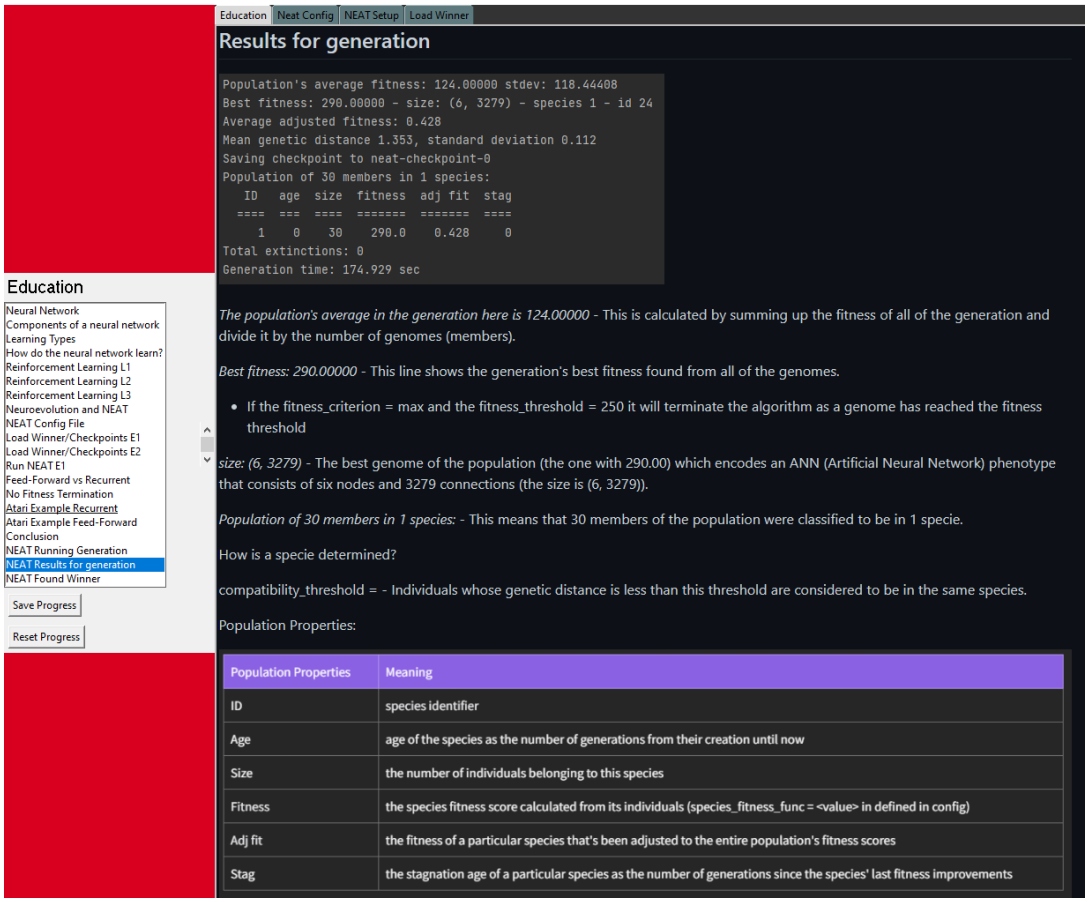
Reset Progress

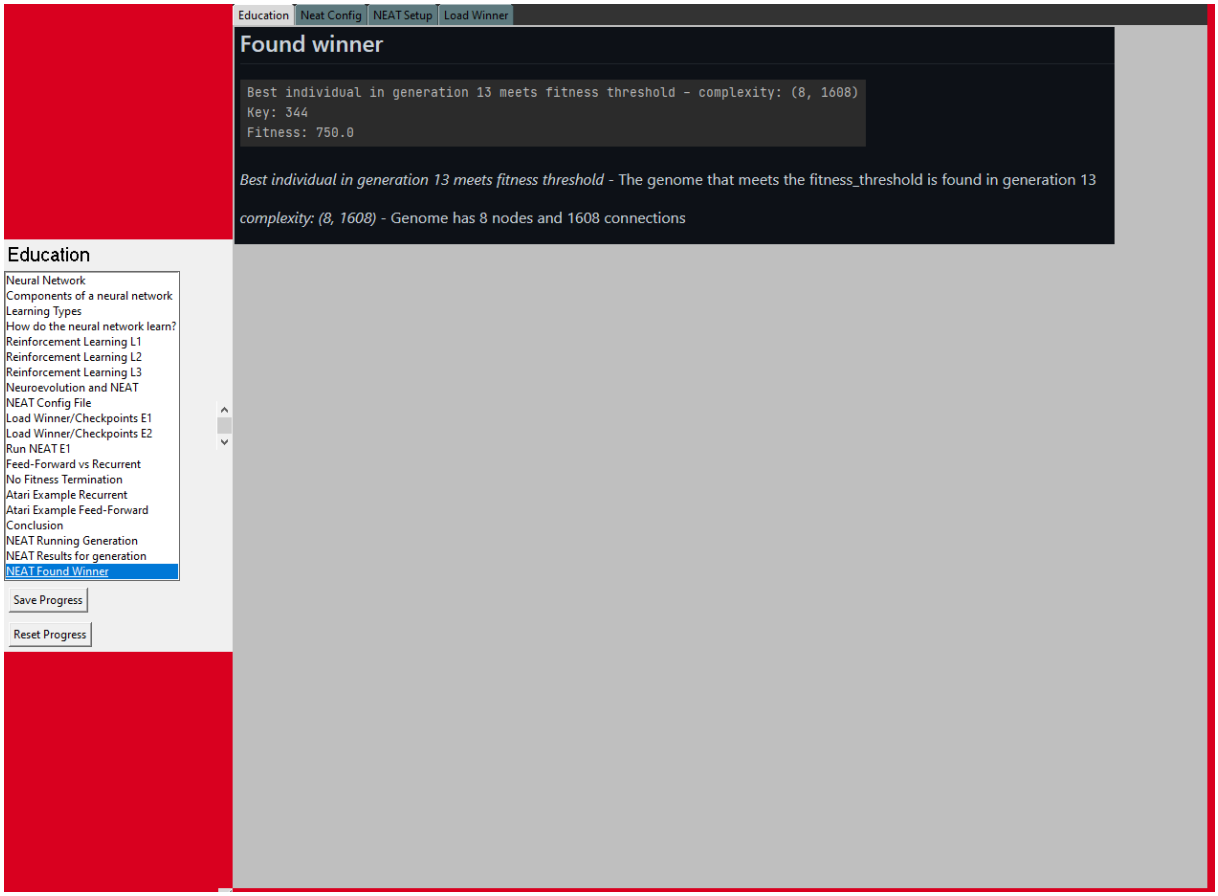

Education
Neat Config
NEAT Setup
Load Winner
The Atari games are simulated though the arcade Learning Environment [ALE], which uses the Stella [Stella] Atari emulator.
You can download the Stella emulator and play the games yourself.
The Atari games are a bit more complicated to play, at least for the computer because it has to process the RGB image and recognise the objects within it. Each input for the games will be 1092 PIXELS and the output will vary depending on the game selected. Every config file can be used with any of the games given you provide correct output.
We will see a Recurrent example first.
Start
Education
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Atari Example Recurrent
Atari Example Feed-Forward
Conclusion
NEAT Running Generation
NEAT Results for generation
NEAT Found Winner
Save Progress
Reset Progress


Education
Neat Config
NEAT Setup
Load Winner
Now that we have seen an example of Recurrent we can see the same example with Feed-Forward for the Atari game Space-Invaders.
Remember the input for each game will be 1092 (pixels) for every Atari game due to the image processing the software is doing to resize the rendered image for the game.
Start
Education
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Atari Example Recurrent
Atari Example Feed-Forward
Conclusion
NEAT Running Generation
NEAT Results for generation
NEAT Found Winner
Save Progress
Reset Progress


Education
Neat Config
NEAT Setup
Load Winner
I hope you have learned some valuable information in the education mode and you can now freely train a neural network and see the results from it. If you would like to see the full guidence for the application you can view it in the ReadMe file.
A few extra notes:
- in order to train your neural network so you reach a threshold, start with a lower threshold in order to view how the score increases in a game and what is the limit
- you can also use no_fitness_termination = True so you can run the game for N generations to get an idea of how the game works
- you can reuse your config file given you change the input and output depending on the game you have chosen
- the games provided in the dropbox are tested games so any other game you try to input will not work
Education
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Atari Example Recurrent
Atari Example Feed-Forward
Conclusion
NEAT Running Generation
NEAT Results for generation
NEAT Found Winner
Save Progress
Reset Progress


Education
Neat Config
NEAT Setup
Load Winner
Running Generation
****** Running generation 0 ******
This line lets the user know which generation is currently the NEAT algorithm at.
Genome and Fitness
****** Running generation 0 ******
Genome: 1 Fitness: 110.0
Genome: 2 Fitness: 0.0
Genome: 3 Fitness: 285.0
Genome: 4 Fitness: 85.0
Genome: 5 Fitness: 0.0
Genome: 6 Fitness: 135.0
The genome is the member of the population. For example if you specify in your Config file that the population will have 30 members you will see that in the 0 generation 30 genomes will be trained (evaluated). If afterwards the number changes that is because NEAT decides if it will remove any genomes or add new ones (from mutation or crossover).
Education
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Atari Example Recurrent
Atari Example Feed-Forward
Conclusion
NEAT Running Generation
NEAT Results for generation
NEAT Found Winner
Save Progress
Reset Progress


Education
Neat Config
NEAT Setup
Load Winner
Results for generation
Population's average fitness: 124.00000 stdev: 118.44408
Best fitness: 290.00000 - size: (6, 3279) - species 1 - id 24
Average adjusted fitness: 0.428
Mean genetic distance 1.353, standard deviation 0.112
Saving checkpoint to neat-checkpoint-0
Population of 30 members in 1 species:
ID age size fitness adj fit stag
1 0 30 290.0 0.428 0
Total extinctions: 0
Generation time: 174.929 sec
Education
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Atari Example Recurrent
Atari Example Feed-Forward
Conclusion
NEAT Running Generation
NEAT Results for generation
NEAT Found Winner
Save Progress
Reset Progress
The population's average in the generation here is 124.00000 - This is calculated by summing up the fitness of all of the generation and divide it by the number of genomes (members).
Best fitness: 290.00000 - This line shows the generation's best fitness found from all of the genomes.
If the fitness_criterion = max and the fitness_threshold = 250 it will terminate the algorithm as a genome has reached the fitness threshold
size: (6, 3279) - The best genome of the population (the one with 290.00) which encodes an ANN (Artificial Neural Network) phenotype that consists of six nodes and 3279 connections (the size is (6, 3279)).
Population of 30 members in 1 species: - This means that 30 members of the population were classified to be in 1 specie.
How is a specie determined?
compatibility_threshold = - Individuals whose genetic distance is less than this threshold are considered to be in the same species.
Population Properties:
Population Properties
Meaning
ID
species identifier
Age
age of the species as the number of generations from their creation until now
Size
the number of individuals belonging to this species
Fitness
the species fitness score calculated from its individuals (species_fitness_func = <value> in defined in config)
Adj fit
the fitness of a particular species that's been adjusted to the entire population's fitness scores
Stag
the stagnation age of a particular species as the number of generations since the species' last fitness improvements


Education
Neat Config
NEAT Setup
Load Winner
Found winner
Best individual in generation 13 meets fitness threshold - complexity: (8, 1608)
Key: 344
Fitness: 750.0
Best individual in generation 13 meets fitness threshold - The genome that meets the fitness_threshold is found in generation 13
complexity: (8, 1608) - Genome has 8 nodes and 1608 connections
Education
Neural Network
Components of a neural network
Learning Types
How do the neural network learn?
Reinforcement Learning L1
Reinforcement Learning L2
Reinforcement Learning L3
Neuroevolution and NEAT
NEAT Config File
Load Winner/Checkpoints E1
Load Winner/Checkpoints E2
Run NEAT E1
Feed-Forward vs Recurrent
No Fitness Termination
Atari Example Recurrent
Atari Example Feed-Forward
Conclusion
NEAT Running Generation
NEAT Results for generation
NEAT Found Winner
Save Progress
Reset Progress

11.3 Survey

11.3.1 Ethics Form

https://livenapierac-my.sharepoint.com/:w:/g/personal/40313507_live_napier_ac_uk/EU0atJalRD1JlfK50VV1kSMBfXP34bLihhzEtQJFpES90w?e=9mJuvT – link to document

Application for Cross-University Ethical Approval

1. Research Details

Name:	Venetsia Tsvetomirova Krasteva
School or Professional service department:	Edinburgh Napier University Merchiston Campus
Email:	40313507@live.napier.ac.uk/venetsia@icloud.com
Contact number:	+447378320249
Project Title:	A graphical application to simplify the use of NEAT to evolve video game characters
Start Date:	18 January
Duration of Project:	12 Weeks
Type of Research: UG/Taught PG/Masters/Doctoral Student/ Staff	

2. Screening Questions

Please answer the following questions to identify the level of risk in the proposed project:
If you answer 'No' to all questions, please complete Section 3a only.
If you have answered 'Yes' to any of the questions 5-14 please complete Section 3a and 3b.
If you have answered 'Yes to any of the questions 1-4, complete all of Section 3.

	You Must Answer All Questions	Yes	No
1.	Is the research clinical in nature?	☐	☒
2.	Is the research investigating socially or culturally 'controversial' topics (for example pornography, extremist politics, or illegal activities)?	☐	☒
3.	Will any covert research method be used?	☐	☒
4.	Will the research involve deliberately misleading participants (deception) in any way?	☐	☒
5.	Does the Research involve staff or students within the University?	☒	☐
6.	Does the Research involve vulnerable people? (For example people under 18 or over 70 years of age, disabled (either physically or mentally), those with learning difficulties, people in custody, migrants etc).	☐	☒
7.	Is the information gathered from participants of a sensitive or personal nature?	☐	☒
8.	Is there any realistic risk of any participants experiencing either physical or psychological distress or discomfort?	☐	☒
9.	Have you identified any potential risks to the researcher in carrying out the research? (for example physical/emotional/social/economic risks?)	☐	☒
10.	Are there implications from a current or previous professional relationship i.e. staff/student/line manager/managerial position that would affect the voluntary nature of the participation?	☐	☒
11.	Will the research require the use of assumed consent rather than informed consent? (For example when it may be impossible to obtain informed consent due to the setting for the research – e.g. observational studies/videoing/photography within a public space)	☐	☒
12.	Is there any risk to respondents' anonymity in any report/thesis/publication from the research, even if real names are not used?	☐	☒
13.	Will any payment or reward be made to participants, beyond reimbursement or out-of-pocket expenses?	☐	☒
14.	Does the research require external ethics clearance? (For example from the NHS or another institution)	☐	☒
15.	Does the research involve the use of secondary data?	☐	☒

3A. Details of Project

In this section please provide details of your project and outline data collection methods, how participant consent will be given as well as details of storage and dissemination.

Please give a 300 word overview of the research project	
The project is on the evolutionary algorithm NEAT and it attempts to make its usage more user friendly providing an easy to use GUI where you can edit the configuration file, train your computer to play a game that the user can select from tested games and load the results afterwards. For people who are new to Artificial Intelligence and would like to learn more and specifically how does it relate to the NEAT algorithm the application includes an educational mode where people can learn more about Artificial Intelligence and be guided on how to use the algorithm.	
Data Collection	
1.	**Who will be the participants in the research?**
	Students and people interested in learning more about AI an in particular the NEAT algorithm (reinforcement learning algorithm) and how to use it.
2.	**How will you collect and analyse the research data? (please outline all methods e.g. questionnaires/focus groups/internet searches/literature searches/interviews/observation)**
	Questionnaires/focus groups/internet search/literature searches/interviews
3.	**Where will the data will be gathered (e.g. in the classroom/on the street/telephone/on-line)**
	online
4.	**Please describe your selection criteria for inclusion of participants in the study**
	Participants will most likely be students in Napier and any person who would like to try the application to train a neural network to play a game
5.	**If your research is based on secondary data, please outline the source, validity and reliability of the data set**
	No it is not
Consent and Participant Information	
7.	**How will you invite research participants to take part in the study? (e.g. letter/email/asked in lecture)**
	letter/email/asked in lecture (AI lecture for example)
8.	**How will you explain the nature and purpose of the research to participants?**
	The purpose of this project is to make the NEAT algorithm more easy to use to train a neural network on games from the Gym environment. It is supposed to be an educational tool where students/people interested in AI can learn new things and try training a neural network.
9.	**How will you record obtaining informed consent from your participants?**
	online
Data storage and Dissemination	

10.	How and in what format will data be stored? And what steps will be taken to ensure data is stored securely?
	It will be stored securely in my university OneDrive account. Authentication is needed to access university accounts and further more the folder can be password protected. In addition only people from the same organisation (the university) can access one's OneDrive provided it is shared or they get the link.
11.	Who will have access to the data?
	Only me and based on the data I will write an overview for evaluation
12.	Will the data be anonymised so that files contain no information that could be linked to any participant?
	The data will be anonymous
13.	How long will the data be kept?
	Until I receive my results for my dissertation (Honours Project), submission is on 1st of December
14.	What will be done with the data at the end of the project?
	It will be deleted securely
15.	How will the findings be disseminated?
	Based on the findings I will draw a conclusion for the evaluation of the application
16.	Will any individual be identifiable in the findings?
	no

3B. Identification and Mitigation of Potential risks

This section is designed to identify any realistic risks to the participants and how you propose to deal with it.

1. **Does this research project involve working with potentially vulnerable individuals?**

Group	Yes	NO	Details (for example programme student enrolled on, or details of children's age/care situation, disability)
Students at Napier	☒	☐	
Staff at ENU	☒	☐	
Children under 18	☐	☒	
Elderly (over 70)	☐	☒	
Disabled	☐	☒	
Migrant workers	☐	☒	
Prisoners / people in custody	☐	☒	
Learning difficulties	☐	☒	

2. **If you are recruiting children (under 18 years) or people who are otherwise unable to give informed consent, please give full details of how you will obtain consent from parents, guardians, carers etc.**

3. **Please describe any identified risks to participants or the researcher as a result of this research being carried out**

There are no risks

4. **Please describe what steps have been taken to reduce these identified risks? (for example providing contact details for appropriate support services (e.g. University Counselling, Samaritans), reminding participants of their right to withdraw and/or not answering questions, or providing a full debriefing to participants)**

There are no risks

5. **If you plan to use assumed consent rather than informed consent please outline why this is necessary**

I do not plan to use assumed consent

6. **If payment or reward will be made to participants please justify that the amount and type are appropriate (for example the amount should not be so high that participants would be financially coerced into taking part, or that the type of reward is appropriate to the research topic).**

There will be no reward or payment

3C. Justification of High Risk Projects

If you answered 'Yes' to the screening questions 1-4 this section asks for justification on the choice of research topic and methodology.

1. **If you have answered yes to question 1 please give a full description of all medical procedures to be used within the research and provide evidence that the project has obtained NHS ethical approval.**

Does not apply

2. **If you have answered yes to questions 2 (research into a controversial topic) please provide a justification for your choice of research topic, and describe how you would deal with any potential issues arising from researching that topic.**

Does not apply

3. **If you have answered yes to questions 3 or 4 (use of deception or covert research methods) please provide a justification for your choice of methodology, and state how you will mitigate the risks associated with these approaches.**

Does not apply

Declaration	
☒	I consider that this project has no significant ethical implications to be brought to the attention of Research Integrity Committee
☐	I consider that this project may have significant ethical implications to be brought to the attention of the Research Integrity Committee
Researcher Signature:	**Date:25/10/2021**
Director of Studies/Supervisor/Principal Investigator Signature:	**Date:**

Checklist

All applications require the following to be submitted with the application form

Participant Information Sheet	☒
Informed Consent Form	☒
Interview/Survey Questions	☒

11.3.2 Survey Consent Form Part one

Feedback from easy to use NEAT graphical user interface application

Page 1 of 4

1. **Consent Form**

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview

You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

☐ **By ticking this box, I acknowledge that I have read and understand**

2. **Consent Form Part I (continued)**

What have you been asked to do?

You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data

For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Would you wish to continue?

☐ **By ticking this box, I acknowledge that I have read and understand**

11.3.3 Survey Consent Form Part two

Feedback from easy to use NEAT graphical user interface application

Page 2 of 4

1. Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.

2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

☐ **By ticking this box, I acknowledge that I have read and understand**

2.

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.

4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**

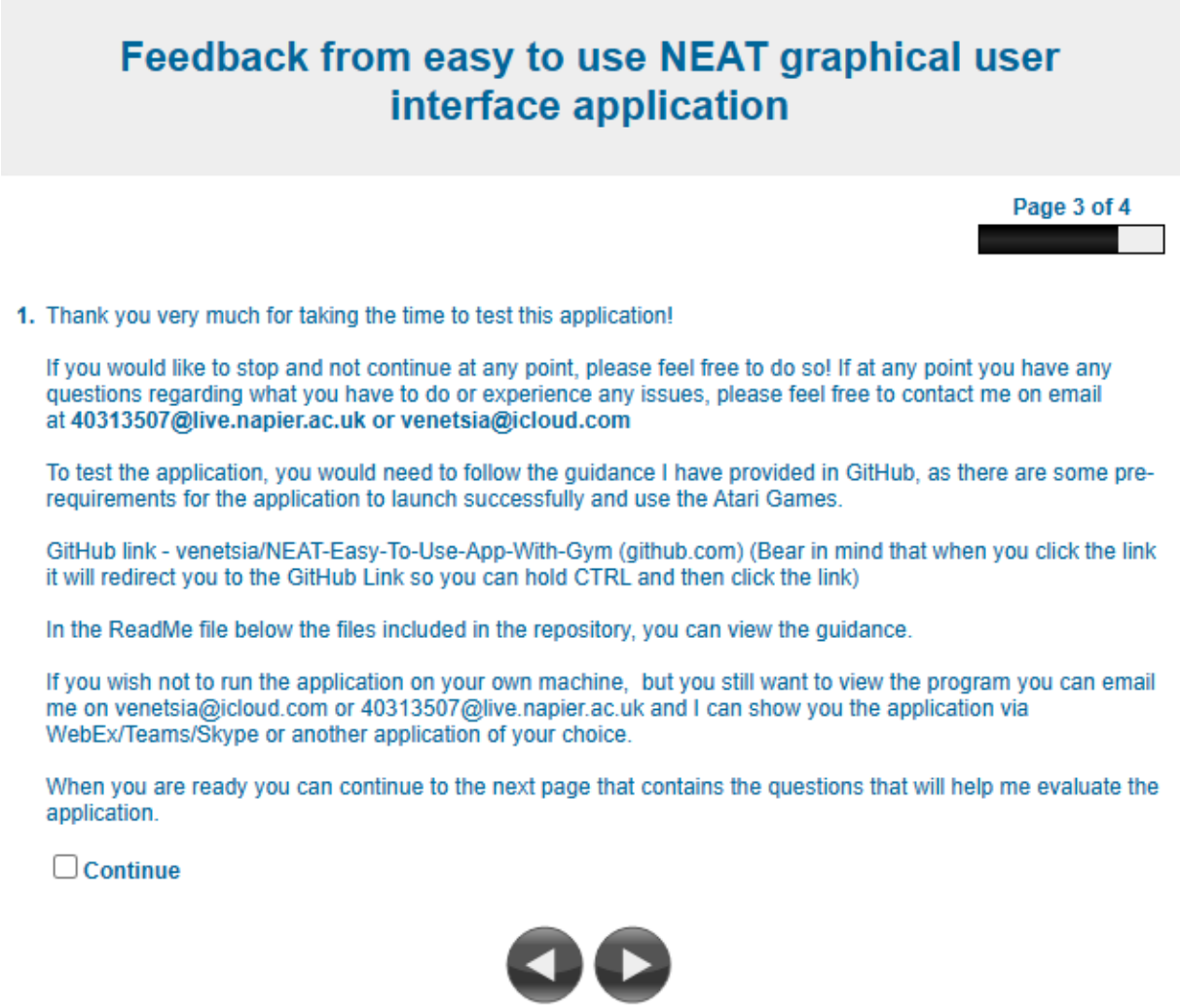
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.

6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.

7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

☐ **By ticking this box, I declare that I have read and understand**

11.3.4 Survey instructions for application



Feedback from easy to use NEAT graphical user interface application

Page 3 of 4

1. Thank you very much for taking the time to test this application!

   If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk or venetsia@icloud.com**

   To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

   GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

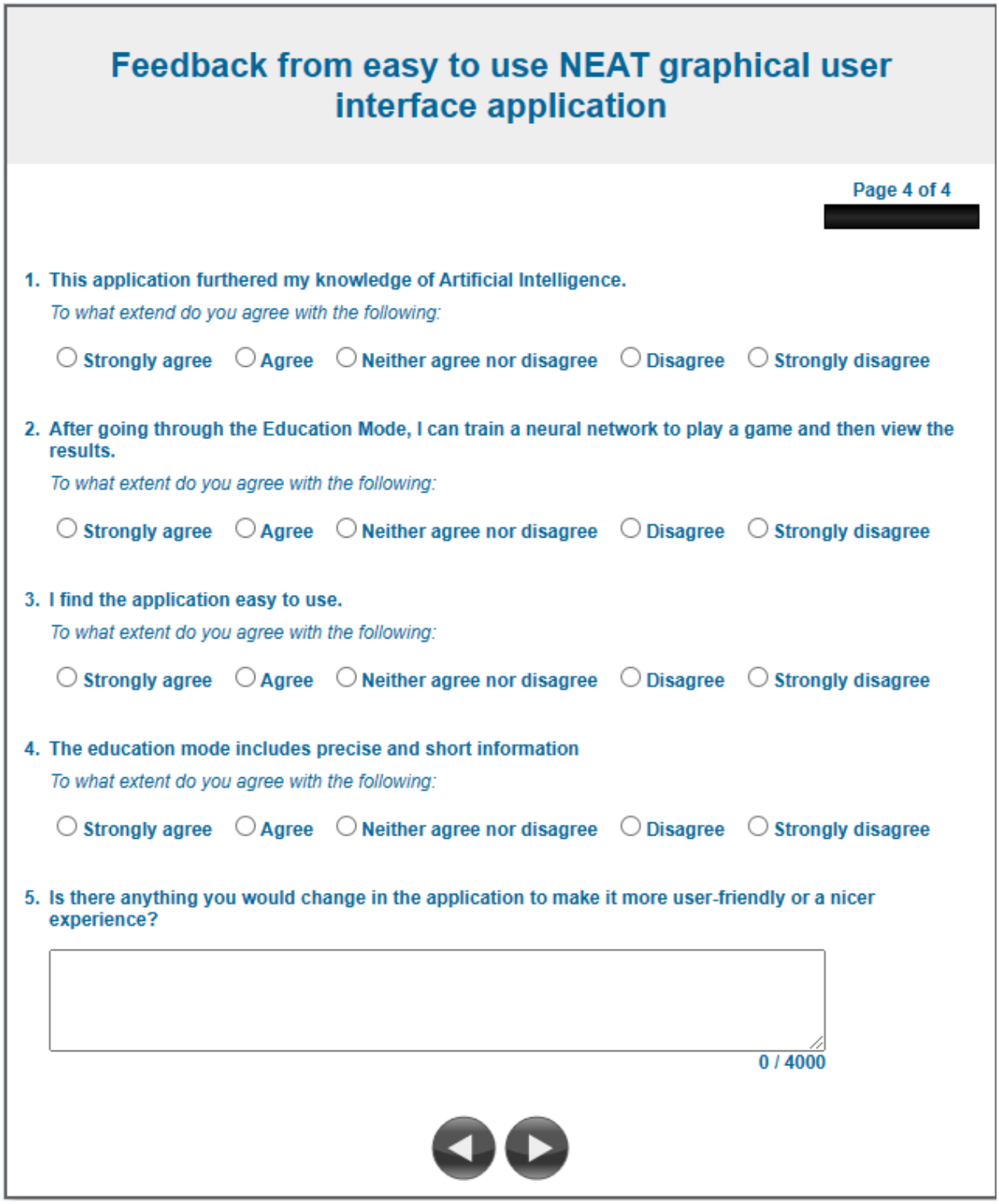
   In the ReadMe file below the files included in the repository, you can view the guidance.

   If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

   When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

   ☐ Continue

11.3.5 Survey Questions



Feedback from easy to use NEAT graphical user interface application

Page 4 of 4

1. This application furthered my knowledge of Artificial Intelligence.
   To what extend do you agree with the following:
   ○ Strongly agree ○ Agree ○ Neither agree nor disagree ○ Disagree ○ Strongly disagree

2. After going through the Education Mode, I can train a neural network to play a game and then view the results.
   To what extent do you agree with the following:
   ○ Strongly agree ○ Agree ○ Neither agree nor disagree ○ Disagree ○ Strongly disagree

3. I find the application easy to use.
   To what extent do you agree with the following:
   ○ Strongly agree ○ Agree ○ Neither agree nor disagree ○ Disagree ○ Strongly disagree

4. The education mode includes precise and short information
   To what extent do you agree with the following:
   ○ Strongly agree ○ Agree ○ Neither agree nor disagree ○ Disagree ○ Strongly disagree

5. Is there anything you would change in the application to make it more user-friendly or a nicer experience?

0 / 4000

11.3.5.1 Questions used in survey:

1. This application furthered my knowledge of Artificial Intelligence.
2. After going through the Education Mode, I can train a neural network to play a game and then view the results.
3. I find the application easy to use.
4. The education mode includes precise and short information.
5. Is there anything you would change in the application to make it more user-friendly or a nicer experience?

11.3.6 Responses

11.3.6.1 First Response

Response of "Anonymous" to survey "Feedback from easy to use NEAT graphical user interface application "

Respondent

Date started 11/08/2021 15:02:30 **Date completed** 11/08/2021 15:03:26 **Time (min)** 0.9

IP address **Language** en-US **Response ID** 8c2cf24f5b4f4bab8943631e4e44918f

Undefined

P1Q1 –

Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview

You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –

Consent Form Part I (continued)

What have you been asked to do?

You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data

For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Should you wish to continue?

By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –

Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Strongly agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Strongly agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Strongly agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Strongly agree

11.3.6.2 Second Response

Response of "Anonymous" to survey "Feedback from easy to use NEAT graphical user interface application "

Respondent

Date started	11/08/2021 16:00:04	**Date completed**	11/08/2021 16:08:15	**Time [min]**	8.3
IP address		**Language**	en-US	**Response ID**	fdc0514769cc404ab287877a0812b024

Undefined

P1Q1 –
Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview
You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –
Consent Form Part I (continued)
What have you been asked to do?
You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data
For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Would you wish to continue?
By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –
Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Agree

11.3.6.3 Third Response

Response of 'Anonymous' to survey 'Feedback from easy to use NEAT graphical user interface application '

Respondent

Date started 11/10/2021 13:52:22 **Date completed** 11/10/2021 13:53:08 **Time (min)** 0.8

IP address: **Language** en-US **Response ID** 44bfad938efb4d75a05df2e572050364

Undefined

P1Q1 –
Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview
You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –
Consent Form Part I (continued)
What have you been asked to do?
You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data
For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Would you wish to continue?
By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –
Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Agree

11.3.6.4 Fourth Response

Response of 'Anonymous' to survey 'Feedback from easy to use NEAT graphical user interface application '

Respondent

Date started	11/11/2021 10:53:26	**Date completed**	11/11/2021 10:54:26	**Time (min)**	1.0
IP address		**Language**	en-US	**Response ID**	fca1e29d491240edabea7c987f6aa234

Undefined

P1Q1 –
Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview
You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –
Consent Form Part I (continued)
What have you been asked to do?
You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data
For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Would you wish to continue?
By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –
Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Agree

P4Q5 – Is there anything you would change in the application to make it more user-friendly or a nicer experience?

This seems really cool, it's great you were able to put it together! I took a quick look at the repository and tutorial there. Do you feel that it met your original goals of enabling playing with NEAT without having to code? Is there a quick executive summary of what you can do with it? If you have software like that now, I think an interesting question is the best way to explain very succinctly what it is and what it does so people who otherwise would be intimidated by such a system can understand quickly that it's accessible and interesting to them. Maybe a very short video or summary of some sort would be useful in that way. Anyway, that aside, this seems like great work, thanks for sharing it!

11.3.6.5 Fifth Response

Response of 'Anonymous' to survey 'Feedback from easy to use NEAT graphical user interface application '

Response

Date started	11/11/2021 22:31:04	**Date completed**	11/11/2021 22:32:36	**Time (min)**	1.5
IP address		**Language**	en-US	**Response ID**	1173a36d87464be5aca3f9d52978605b

Undefined

P1Q1 –

Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview

You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –

Consent Form Part I (continued)

What have you been asked to do?

You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data

For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Should you wish to continue?

By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –

Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:

Strongly agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:

Strongly agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:

Agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:

Strongly agree

P4Q5 – Is there anything you would change in the application to make it more user-friendly or a nicer experience?

Venetsia did awesome work!
Everything is well presented,informing and works perfectly!

11.3.6.6 Sixth Response

Response of 'Anonymous' to survey 'Feedback from easy to use NEAT graphical user interface application '

Respondent

Date started	11/22/2021 20:00:07	**Date completed**	11/22/2021 20:00:42	**Time [min]**	0.6
IP address		**Language**	en-US	**Response ID**	69e51c4fe725499cb3276aa6b6f7445d

Undefined

P1Q1 –
Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview
You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –
Consent Form Part I (continued)
What have you been asked to do?
You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data
For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Would you wish to continue?
By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –
Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Strongly agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Strongly agree

11.3.6.7 Seventh Response

Response of 'Anonymous' to survey 'Feedback from easy to use NEAT graphical user interface application '

Respondent

Date started	11/22/2021 20:00:46	**Date completed**	11/22/2021 20:01:07
Time (min)	0.4	**IP address**	
Language	en-US	**Response ID**	37736a2961e0439d801c9c29202b3d54

Undefined

P1Q1 –

Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview

You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –

Consent Form Part I (continued)

What have you been asked to do?

You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data

For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Would you wish to continue?

By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –

Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Strongly agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Strongly agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Strongly agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Strongly agree

11.3.6.8 Eight Response

Response of 'Anonymous' to survey 'Feedback from easy to use NEAT graphical user interface application '

Respondent

Date started	11/22/2021 20:01:11	**Date completed**	11/22/2021 20:01:36	**Time (min)**	0.4
IP address		**Language**	en-US	**Response ID**	48e839ec11f74b2792ee2901949278f3

Undefined

P1Q1 –
Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview
You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –
Consent Form Part I (continued)
What have you been asked to do?
You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data
For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Should you wish to continue?
By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –
Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Strongly agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Strongly agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Strongly agree

11.3.6.9 Ninth Response

Response of 'Anonymous' to survey 'Feedback from easy to use NEAT graphical user interface application '

Response

Date started	11/22/2021 20:02:08	**Date completed**	11/22/2021 20:02:31	**Time (min)**	0.4
IP address		**Language**	en-US	**Response ID**	f344783945124c53a310bdb672ec09e6

Undefined

P1Q1 –
Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview
You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –
Consent Form Part I (continued)
What have you been asked to do?
You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data
For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Should you wish to continue?
By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –
Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.
To what extend do you agree with the following:
Strongly agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.
To what extent do you agree with the following:
Agree

P4Q3 – I find the application easy to use.
To what extent do you agree with the following:
Agree

P4Q4 – The education mode includes precise and short information
To what extent do you agree with the following:
Strongly agree

11.3.6.10 Tenth Response

Response of 'Anonymous' to survey 'Feedback from easy to use NEAT graphical user interface application '

Response

Date started	11/22/2021 20:02:36	**Date completed**	11/22/2021 20:03:05	**Time [min]**	0.5
IP address		**Language**	en-US	**Response ID**	515786eab6f9410cb6c2ee3527ef6862

Undefined

P1Q1 –

Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk

Overview

You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –

Consent Form Part I (continued)

What have you been asked to do?

You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data

For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Would you wish to continue?

By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –

Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Strongly agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Strongly agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Strongly agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Agree

11.3.6.11 Eleventh Response

Response of 'Anonymous' to survey 'Feedback from easy to use NEAT graphical user interface application '

Response

Date started	11/22/2021 20:03:09	**Date completed**	11/22/2021 20:03:22	**Time (min)**	0.2
IP address		**Language**	en-US	**Response ID**	b331bb4358f24b6b88216a899dabe8d9

Undefined

P1Q1 –

Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview

You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –

Consent Form Part I (continued)

What have you been asked to do?

You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data

For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Should you wish to continue?

By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –

Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Strongly agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Strongly agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Strongly agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Strongly agree

11.3.6.12 Twelfth Response

Response of 'Anonymous' to survey 'Feedback form easy to use DENT graphical user interface application '

Response

Date started	11/22/2021 20:03:26	**Date completed**	11/22/2021 20:03:42	**Time (min)**	0.3
IP address		**Language**	en-US	**Response ID**	3e24fbfc804448e29b1793edfdd48df0

Undefined

P1Q1 –
Consent Form

Edinburgh Napier UNIVERSITY

Participant information sheet (informed consent part I)

A graphical application to simplify the use of NEAT to evolve video game characters

Dear Participant,

Thank you for taking the time to read through the following information and try out the application. If at any point you have any further questions, please do not hesitate to contact 40313507@live.napier.ac.uk.

Overview
You have been invited to take part in testing the application and giving your opinion, feedback and answering a couple of questions.
The purpose of the application is to further people's knowledge of artificial intelligence and try out the NEAT learning algorithm so you can see what artificial intelligence can achieve. The second purpose of this application is to make the NEAT algorithm easier to use, providing you with a simple graphical user interface. In the graphical user interface, you can edit the configuration file, train your computer to play a game and then view the results so that anyone can use it.

By ticking this box, I acknowledge that I have read and understand

P1Q2 –
Consent Form Part I (continued)
What have you been asked to do?
You have been asked to participate in a confidential and anonymous online survey that concerns the application developed for educational purpose and ease the use of the NEAT algorithm, an artificial intelligence reinforcement learning algorithm.
This participation is on a voluntary basis and at any given time you are free to withdraw from it.
In order to be able to answer the questions ahead you would need to have viewed the applications education mode and normal mode.

Your data
For the purpose of understanding if the application has reached its goal your information will be anonymous and confidential. The results will be used for evaluation.
The answers you submit will only be available to me and will be securely deleted after the results from the project have come out.

Would you wish to continue?
By ticking this box, I acknowledge that I have read and understand

Undefined

P2Q1 –

Consent Part II

Edinburgh Napier UNIVERSITY

Informed Consent Form (informed consent part II)

A graphical application to simplify the use of NEAT to evolve video game characters

Edinburgh Napier University requires that all persons who participate in research studies give their written consent to do so. Please read the following and sign it if you agree with what it says.

1. I freely and voluntarily consent to be a participant in this research to be conducted by Venetsia Tsvetomirova Krasteva, who is an undergraduate/ postgraduate student / staff member in the Edinburgh Napier School of Computing.
2. I have been informed of the broad goal of this research study. I have been told what is expected of me and that the study should take no longer than 30 minutes to complete.

By ticking this box, I acknowledge that I have read and understand

P2Q2 –

3. I have been told that my responses will be anonymised. My name will not be linked with the research materials, and I will not be identified or identifiable in any report subsequently produced by the researcher. I have been told that these data are may be submitted for publication.
4. **I also understand that if at any time during the survey I feel unable or unwilling to continue, I am free to leave. That is, my participation in this study is completely voluntary, and I may withdraw from it at any time without negative consequences.**
5. In addition, should I not wish to answer any particular question or questions, I am free to decline.
6. I have been given the opportunity to ask questions regarding the survey and my questions have been answered to my satisfaction.
7. I have read and understand the above and consent to participate in this study. My signature is not a waiver of any legal rights. Furthermore, I understand that I will be able to keep a copy of this consent form for my records.

By ticking this box, I declare that I have read and understand

Undefined

P3Q1 –
Thank you very much for taking the time to test this application!

If you would like to stop and not continue at any point, please feel free to do so! If at any point you have any questions regarding what you have to do or experience any issues, please feel free to contact me on email at **40313507@live.napier.ac.uk** or **venetsia@icloud.com**

To test the application, you would need to follow the guidance I have provided in GitHub, as there are some pre-requirements for the application to launch successfully and use the Atari Games.

GitHub link - venetsia/NEAT-Easy-To-Use-App-With-Gym (github.com) (Bear in mind that when you click the link it will redirect you to the GitHub Link so you can hold CTRL and then click the link)

In the ReadMe file below the files included in the repository, you can view the guidance.

If you wish not to run the application on your own machine, but you still want to view the program you can email me on venetsia@icloud.com or 40313507@live.napier.ac.uk and I can show you the application via WebEx/Teams/Skype or another application of your choice.

When you are ready you can continue to the next page that contains the questions that will help me evaluate the application.

Continue

Undefined

P4Q1 – This application furthered my knowledge of Artificial Intelligence.

To what extend do you agree with the following:
Agree

P4Q2 – After going through the Education Mode, I can train a neural network to play a game and then view the results.

To what extent do you agree with the following:
Agree

P4Q3 – I find the application easy to use.

To what extent do you agree with the following:
Agree

P4Q4 – The education mode includes precise and short information

To what extent do you agree with the following:
Agree

11.4 Contact with Kenneth Stanley:

https://livenapierac-my.sharepoint.com/:f:/r/personal/40313507_live_napier_ac_uk/Documents/Dissertation%20Project%20Management/Kenneth%20Stanley%20email?csf=1&web=1&e=0tQP2F

11.5 Contacting Babis Konairis:

https://livenapierac-my.sharepoint.com/:f:/r/personal/40313507_live_napier_ac_uk/Documents/Dissertation%20Project%20Management/Babis%20Koniaris?csf=1&web=1&e=rMdOfY