

Deliverable 1: Final Year Dissertation

Evolving a Learning Agent using
Neuroevolution in the FightingICE Game
Platform

Robert John Dunn
H00163867
BSc Honours in Computer Science
Heriot-Watt University

Supervisor:
Dr Patricia A. Vargas

Co-Supervisor:
Dr Fabrício Olivetti de França

Second Reader:
Dr Mohamed Abdelshafy

Declaration

I, Robert Dunn confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:

Date:

Abstract

Neuroevolution is a popular technique for machine learning in which an artificial neural network is trained by an evolutionary algorithm. The technique takes inspiration from the evolution of the biological nervous system and is a popular approach for reinforcement learning problems. One way to demonstrate the effectiveness of neuroevolution is through artificial intelligence in games. This project aims to implement a learning agent in the FightingICE platform, a two-dimensional Java fighting game organised and maintained by Ritsumeikan University, Kyoto. The agent is designed to evolve through neuroevolution to improve its performance in the game, eventually becoming competitive versus a human opponent. By implementing a neuroevolution method in a simplistic environment, we hope to evaluate the effectiveness of neuroevolution as a method of machine learning and explore the potential of our agent's performance.

Contents

1	Introduction	5
2	Literature Review	6
2.1	Machine Learning	6
2.1.1	Learning Paradigms	6
2.1.2	Incremental Evolution	6
2.2	Neuroevolution	7
2.2.1	Fundamentals	7
2.2.2	Artificial Neural Networks	7
2.2.3	Genetic Algorithms	8
2.2.4	Neuroevolution in Games	8
2.3	FightingICE	9
2.3.1	Game Platform	9
2.3.2	Game Agents	9
2.3.3	Related Projects	10
3	Requirements	11
4	Evaluation Strategy	12
5	Project Management	13
6	References	13
7	Appendices	13

1 Introduction

Machine learning is a heavily researched field of artificial intelligence in which machines are given the ability to learn without being explicitly programmed. Various methods of machine learning exist which allow the machine to adapt itself in order to process new data. One of these methods is neuroevolution, which involves an artificial network being trained with an evolutionary algorithm to imitate the learning process.

The neuroevolution method is loosely based on the way the biological nervous system operates: with neurons communicating through axons, represented by nodes of the neural network with weighted connections. Neuroevolution is a form of reinforcement learning, which means a notion of reward is introduced and the machine attempts to maximise said reward. The wide application of reinforcement has given neuroevolution popularity in the fields of artificial life and evolutionary robotics. This notion of reward also allows the method to easily be applied to computer games since the reward is usually simple to measure, e.g. the hit-points of the player's character.

This project aims to implement a neuroevolution algorithm acting on a learning agent in the FightingICE game platform. FightingICE is a two-dimensional fighting game written in Java by Intelligent Computer Entertainment Lab, Japan. The platform allows easy implementation of artificial intelligence by sending delayed game information to the agent. Using neuroevolution we hope to evolve an agent to the point of being competitive versus a human opponent, though not unbeatable. The performance of the agent will be evaluated at various stages of the evolution in order to evaluate the effectiveness of our algorithm.

The next section of this dissertation is the literature review which discusses and evaluates relevant technical literature. Followed by requirements for the project identifying system requirements for the target system. Next, the evaluation strategy for the project is presented which details evaluation and analysis to be conducted. Finally an overview of management for the project which includes a timetable for the project, analysis of risks, and considerations of professional, legal, ethical and social issues.

2 Literature Review

2.1 Machine Learning

2.1.1 Learning Paradigms

Learning is an essential human function in which we modify our behaviour tendency according to experiences, to become better when a similar situation occurs. In the study of machine learning, algorithms, computer applications, and systems, utilise learning to improve their performance at certain tasks. There are two main entities in the machine learning model, the teacher and the learner. The teacher contains the knowledge to perform a given task while the learner has to learn the knowledge the teacher holds. [9] There are three types of machine learning [pat]:

1. *Supervised Learning*

A teacher provides the learner with a set of input and desired output pairs. The learner can then use these examples to improve its performance at the task.

2. *Unsupervised/Self-organised Learning*

There is no teacher, so the learner learns based only on the stimuli received. A common application is finding patterns or grouping within data, using methods such as cluster analysis.

3. *Reinforcement Learning*

The agent uses goal-directed learning where a notion of reward is introduced and the agent attempts to maximise this reward. There is no teacher for the agent and no explicit model of the environment.

2.1.2 Incremental Evolution

When an agent is provided with a complex behaviour, it may have trouble evolving to perform at its potential. Using incremental evolution, the behaviour can be learnt incrementally with tasks gradually increasing in difficulty. This form of evolution proved effective in one implementation [5]. The agent's task was a prey capture task: the agent moves through the environment and must catch its prey before the set number of time-steps. With increasingly difficult tasks, the agent was able to rapidly improve its performance, and skip many potential generations of evolution.

2.2 Neuroevolution

2.2.1 Fundamentals

Neuroevolution is a popular biologically-inspired method of machine learning in which an artificial neural network is evolved using an evolutionary algorithm. The method has proved popular, especially in the fields of artificial life, evolutionary robotics and computer games. One reason for its popularity is the fact that neuroevolution is a form of reinforcement learning, which can be applied more generally than its counterpart, supervised learning.

2.2.2 Artificial Neural Networks

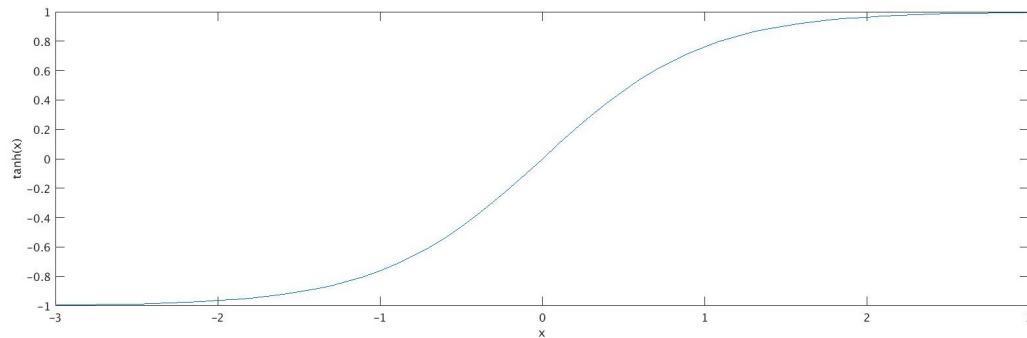
Artificial neural networks model the way the brain solves problems with collections of neurons communicating through axons, represented with neurons with weighted connections. Neural networks have an associative (content-addressable) memory and can also implement parallel processing. They are often applied to pattern finding within data.

Topology

The most basic version of an ANN (artificial neural network) is a perceptron. A perceptron is a simple machine which takes a variable number of input nodes which connect to a single output node. From the perceptron evolved the multi-layer feed forward network, which incorporates extra 'hidden' layers of neurons between the input and output. The more hidden layers there are and the more nodes in these layers, the more complex behaviour a network can experience.

Activation Function

In order to saturate neuron values to a usable range (usually -1 to 1), an activation function is used. The most basic version of this function is ReLU , where any values below zero will result in an activation of zero, and any values above zero in an activation of one. It is also possible to use more general, non-linear functions such as the tanh function, shown below.



Another possibility is to have neurons operate probabilistically.[add more]

NEAT (Neuroevolution of Augmenting Topologies)

TO DO

2.2.3 Genetic Algorithms

2.2.4 Neuroevolution in Games

How neuroevolution can be used to evolve learning agents in games, usually agent is optimising some value e.g. HP. How fitness in games is evaluated, giving some examples. How neuroevolution fares compared to other learning algorithms. Present examples of projects implementing neuroevolution to learn.

2.3 FightingICE

2.3.1 Game Platform

FightingICE is a Java based game platform organised and maintained by Intelligent Computer Entertainment Lab., Ritsumeikan University. The game is based in an arena where two fighters are competing versus each other, attempting to deplete the other's hit-points while preserving their own. The FightingICE platform was designed to allow easy development and evaluation of artificially agents in the game for research or hobby purposes. Once implemented, an agent receives information about the state of the game from the platform periodically, such as the opponent player's location and current energy levels. A delay is added to this game information in order to simulate the delay a human player would experience from reaction time.



2.3.2 Game Agents

Agents in the game have two main attributes- hit-points and energy. There are four characters available in the game: Zen, Garnet, Lud, and Kfm. Each of these characters is capable of moving, performing attacks, and combining these attacks into deadly combos. Present information on the agents in the FightingICE game. The capabilities of the agents including which attack moves / movement commands they can do. Touch on fitness evaluation (how to win the game: keep enemy HP low and your HP high).

2.3.3 Related Projects

Discuss and reference relevant projects which have been completed using the FightingICE framework. Discuss what has been achieved in said projects and the potential further research that can be undertaken.

3 Requirements

Functional:

FR-1 : The agent should be barely competitive versus a human opponent.

FR-2 : A neural network to control the agent's behaviour should be implemented.

FR-3 : An appropriate evolutionary algorithm to evolve the neural network (see FR-2) should be implemented.

FR-4 : The agent's performance versus a human opponent at various stages of evolution should be evaluated.

FR-5 : The agent should be coded in the Java FightingICE platform.

Non-functional:

NFR-1 : Incremental evolution could be utilised to skip some generations of evolution.

NFR-2 : Evolution of the neural network could be visualised.

4 Evaluation Strategy

Evaluation of the agent's performance will be calculated through the character's hit-points attribute. In FightingICE, both players start at zero hit-points and connected attacks cause these hit-points to be reduced below zero. Once the game is finished, comparisons can be made between both player's hit-points to evaluate the performance of the agent.

Evaluation will consist of games played versus a human opponent at different stages of the agent's evolution. This will allow us to analyse how the agent's performance improves over time and in turn the effectiveness of neuroevolution.

[ADD MORE]

5 Project Management

6 References

<http://www.cs.utexas.edu/users/nn/downloads/papers/gomez.ijcai99.pdf>
<https://arxiv.org/abs/1608.02971>
<https://arxiv.org/abs/1604.00644>
<https://arxiv.org/abs/1312.5355>
<https://arxiv.org/abs/1107.0037> <http://nn.cs.utexas.edu/keyword?gomez:ab97>

7 Appendices

References

- [1] Wilde, P (1997) **Neural Network Models** [book] P4-P14
- [2] Risi, S & Togelius, J (2015) **Neuroevolution in Games: State of the Art and Open Challenges** [online] Available at: <https://arxiv.org/pdf/1410.7326.pdf> [Accessed 03 Nov. 2016]
- [3] Pace, A (2014) **Improving AI for simulated cars using Neuroevolution** [online] Available at: <http://commerce3.derby.ac.uk/ojs/index.php/gb/article/view/3/1> [Accessed 07 Nov. 2016]
- [4] Lampropoulos, A (2005) **Machine Learning Paradigms** [online] Available at: <http://file.allitebooks.com/20150722/Machine%20Learning%20Paradigms-%20Applications%20in%20Recommender%20Systems.pdf> [Accessed 18 Nov. 2016]
- [5] Gomez, F & Miikkulainen, R (1997) **Incremental Evolution of Complex General Behaviour** [online] Available at: <http://nn.cs.utexas.edu/downloads/papers/gomez.adaptive-behavior.pdf> [Accessed 18 Nov. 2016]
- [6] Batsford, T (2014) **Calculating Optimal Jungling Routes in DOTA2 Using Neural Networks and Genetic Algorithms** [online] Available at: <http://commerce3.derby.ac.uk/ojs/index.php/gb/article/view/14/12> [Accessed 18 Nov. 2016]

- [7] Hausknecht, M & Lehman, J & Stone, P (2014) **A Neuroevolution Approach to General Atari Game Playing** [online] Available at: <https://www.cs.utexas.edu/~mhauskn/papers/atari.pdf> [Accessed 15 Nov. 2016]
- [8] Braylan, A & Hollenbeck, M & Meyerson, E & Miikulainen, R (2015) **Reuse of Neural Modules for General Video Game Playing** [online] Available at: <https://arxiv.org/pdf/1512.01537v1.pdf> [Accessed 17 Nov. 2016]
- [9] Dehuri, S & Ghosh, S & Cho, S-B (2011) **Integration of Swarm Intelligence and Artificial Neural Network** [book] P1-P23