**NEAT and overcoming local optima**

**through at Drills-based approach**

Joshua Beemster, Jasim Schluter and Loic Nyssen

03/09/2018

Advisor: Jim Tulip

## Abstract

This paper investigates NEAT and the effect on NEATs performance when training is tailored towards a repeated, focused training approach, vs. a general run-through approach that is common in the game AI training area.

This paper also investigates the issue of cross-species mating in terms of its effectiveness in generalising specifically trained Neural Networks that were trained using the Drill approach.

## Introduction

Neural Networks are generalisable decision making systems, Neural networks have been shows to be effective in areas such as: image recognition, voice recognition, image classification, and increasingly in the area of game playing.

Neural Evolution of Augmented Topologies is a genetic algorithm that produces Neural Networks. It was invented by Phd. Kenneth Stanley in 2001. It works by keeping a large population of Neural Networks. Each neural network is started in a sparse, minimalistic way, and via using an evolutionary algorithm, more nodes and weights are added to the network, and those networks are combined and mutated and culled based on the fitness.

While NEAT has been shown to be able to train an agent to play the game Super Mario Bros., it might be true that it takes longer for the Neural network to learn than the most optimal strategy, since playing through games starting from the start of the game does not focus the fitness algorithm as specifically as a tailored algorithm that focuses the neural network on training against the toughest parts of the game.

## Related Work

This paper heavily draws on the research done by Kenneth Stanley. It also draws on the work by Julian Togelius on areas of Game research with interesting open questions and on the work by Uber research on overcoming local optima via novel approaches.

So far we have not see any papers on the effectiveness of a Drill-based approach, nor have we yet found an academic reference to the effectiveness of using cross-species breading to generalise specifically trained Neural Networks.

OpenAI have held competitions to see what researchers can do in the area of training an agent to play games. One notable one is the Sonic competition. These competitions usually require the agent to play through from the start of a mystery level that the agent hadn’t trained on before. It is hoped by doing so the agent will prove it has generalised learning, not simply memorised the level.

## Terminology

* Organism, genome and individual - a single neural network within NEAT
* Species - one of several disjoint groups of similar genomes maintained by NEAT for population management
* Speciation – a collective term for the set of processes by which NEAT creates, maintains and uses species
* Mating – a process by which two existing genomes are combined to form a new genome. The combined genomes are called parents, and the new genome is the child
* Interspecies mating - mating where the two combined genomes are in different species
* Adoption - either an instance of mating where the child is in a different species from either of its parents or the process by which these adoptions occur
* Episode – a training run that when it’s fitness score is used as part of a weighted average of the fitness score of a genome results in a greater statistical significate measure of the fitness of the genome.
* Drill-based – a process of attaining a new skill by focusing training on a repeating a specific desired skill. Often used in sports training.
* Stuck points – local optima in the multi-dimensional state space of potential solutions to competing the level. The agent may only be stuck for a limited number of generations; however they result in impedance of the discovery of the global optimal solution.

## Experimental Setup

The setup was NEAT-Python version 0.92. The Open AI gym environment was version 0.10.5. We forked a Super Mario gym found on github at: https://github.com/ppaquette/gym-super-mario/. The version of the FCEUX emulator was built from the latest source code as of commit number 4939859: https://github.com/TASVideos/fceux/commit/4939859cf466530e7e5771b27074496ea2c605f0 as this had a feature of allowing saving and loading of game states via api, that was required for focusing training on a specific area. Experiments were run in Docker containers, with an example agent being Agent.py in the NEAT folder. All code is checked into GitHub at: <https://github.com/NES-NN/OpenAI-Testbed>

The task being evaluated was how rapidly is the agent playing Mario in SMB level 1 is able to pass through the level. We found by trial and error with the algorithm NEAT that Mario was getting stuck at points in the game where new behaviours needed to be learnt for Mario to pass through that point in the game.

These are shown in the figure below:

[Mario getting stuck at low jumping point]

[Mario getting stuck at high jumping point]



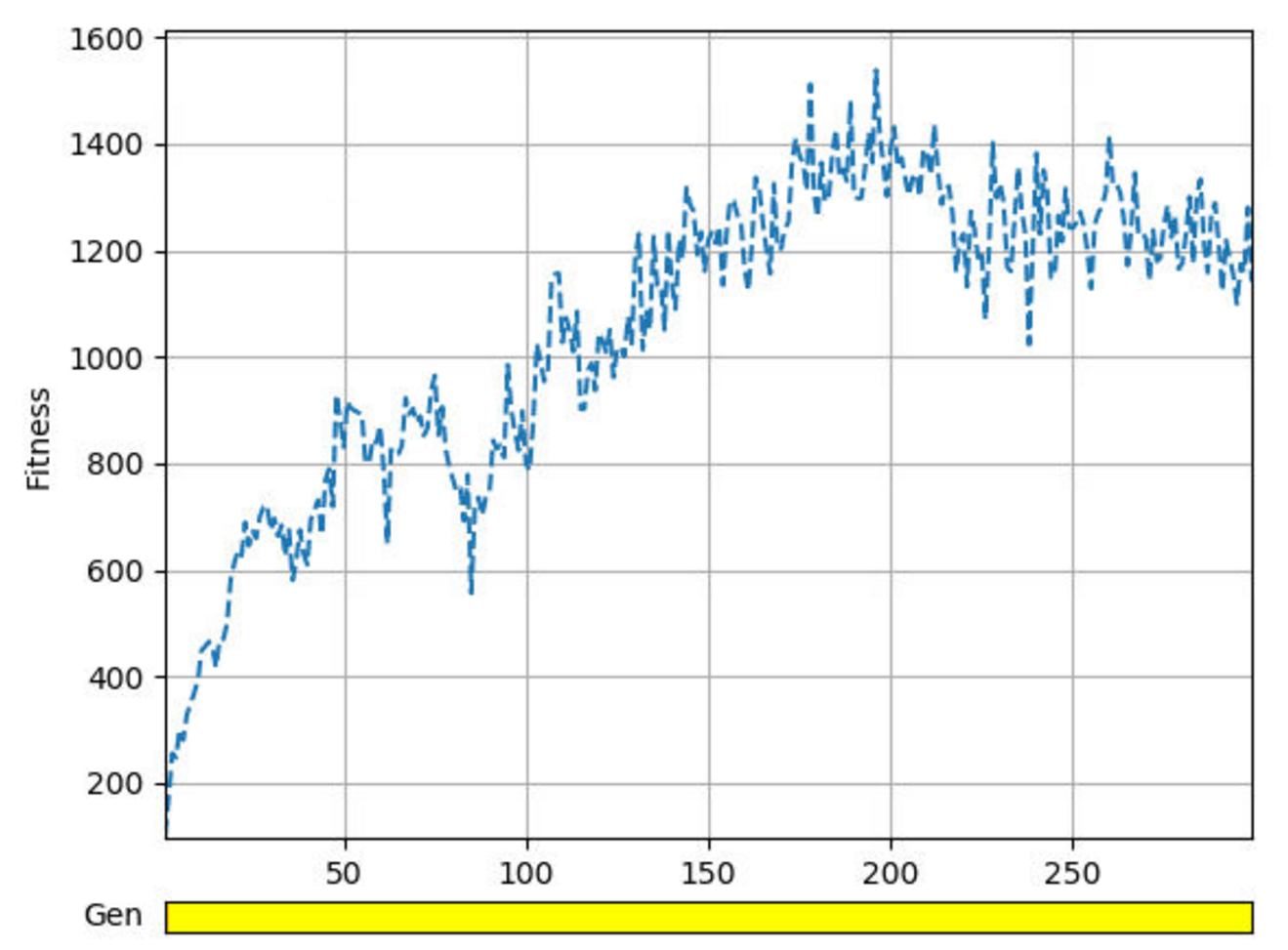
[Mario getting stuck at hole]

[Mario getting stuck at stair step]



[Mario getting stuck in water]

As these points inhibited the growth of fitness of the Genomes, see figure 2:



[Figure 2: History of genome growth over Generations 400]

This paper investigated specifically the question, “what would happen if we focused training specifically on those areas that the network had the hardest time on learning, adapting for, or otherwise trying to overcome, by using a focused drill-like approach.

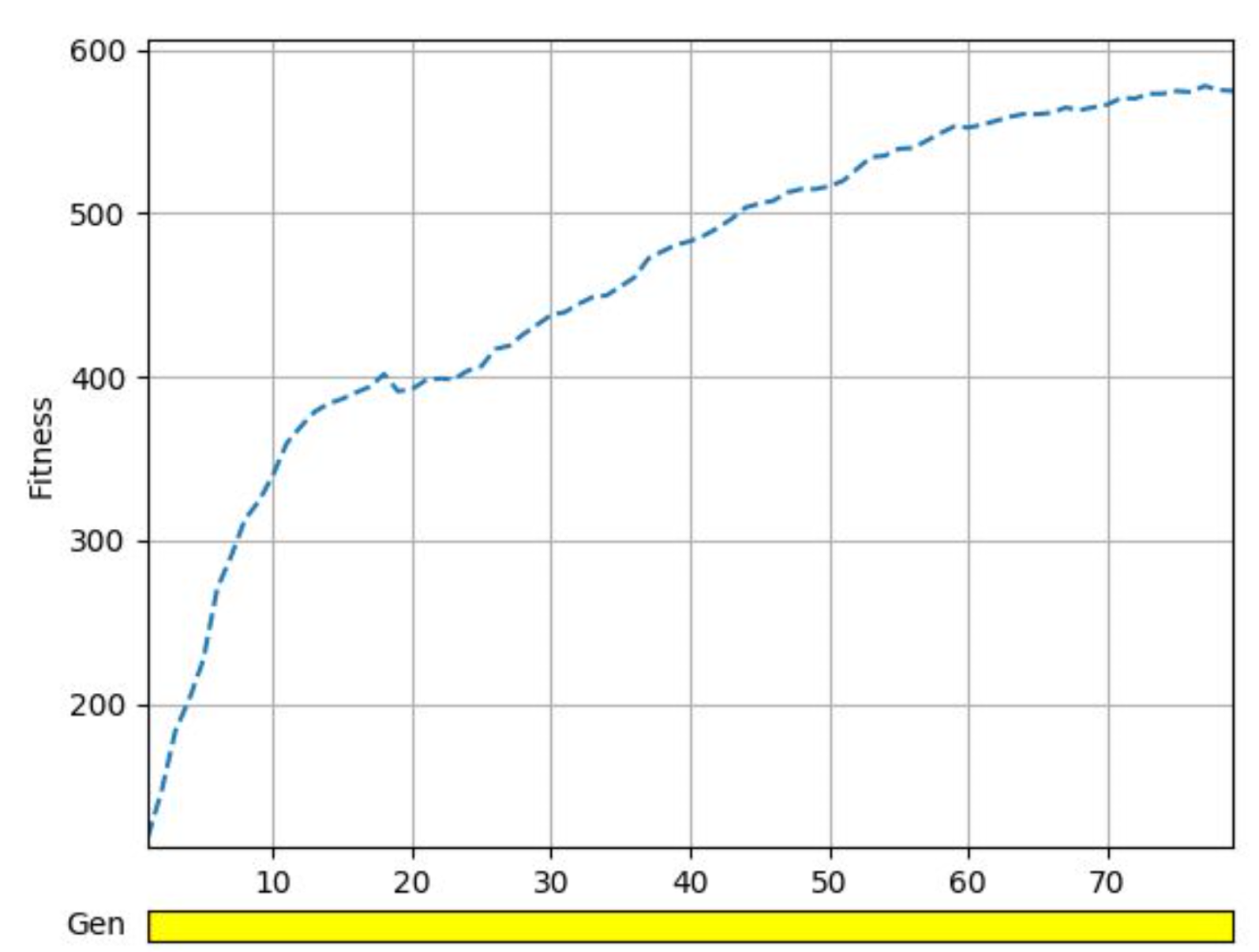
Experiments were done over a range of 20 genomes, 100 genomes, 400 genomes, with multiple parallel trainings and serial training modes.

## Experiments

This section explains several individual experiments that build towards the eventual conclusion. The next section compares the results of those experiments.

### Baseline Experiment

Mario was trained using the NEAT approach through 400 generations to uncover how long it would take and what were the stuck points. These are shown in the figure below:



[Figure 3: Fitness over Generations for 80 runs]

We noted that there was a slowdown in location x, y, z and this can be shown with formula:

Std dev. / current run – previous run

And is quite clear in the following chart:

[Figure 4: fitness over generations chart with trend lines added to highlight the slowing growth]

### Experiment 2

Running Mario across multiple stuck points as part of a training episodes.

Here Mario is teleported to the different areas that he gets stuck, and is forced to train against all of those, taking the overall fitness score increase in each area as the weighted score for judging the overall fitness of the genome.

Here we see that while he takes a lot longer to evolve, the rate of fitness increase is steadier, and more linear, and the fitness score continues to increase higher. And given 500 generations this shows that a more generalised neural network has evolved that what otherwise develops in NEAT runs.

[Figure 4: Fitness over generation chart showing slow but steady growth]

### Experiment 3.

Here we take a Neural network that has been trained at each of the stuck points specifically though a 100 generations each, and then turn on inter-species mating in order to attempt to mix these genomes together in order to form a neural network that is more generalised fit algorithm.

This mating was encouraged due to configuration that was set in the NEAT algorithm and the results are shown in the below diagram:

[Figure 5: fitness over generation of pre-trained genomes with high inter-species breading on.]

### Experiment 4

Jumping Mario to random points in the level and having him train on them.

This experiment seeks to confirm that the benefit that the algorithm is producing is not from simply training in different locations, that the information density (new novel challenge) of the stuck points is higher and therefor the neural networks learn the most from being trained on those points.

### Experiment 5.

We are considering running the trained agent against a level that he was not trained for. This is to demonstrate that the drills approach results in skills that are reused, rather than simply memorisation of the level.

### Experiment 6.

Deep NEAT. There is potential to train a neural network to supervise the agent, switching between the specifically trained networks as the level progresses. This is likely to become a stretch goal of the paper.

## Summary and Discussion of results

It is currently too early to predict the results of these experiments. The experimenters are hoping to show that there is a faster recovery from the areas that usually slow the fitness rate. That the detection of the stuck points provided useful input into an optimisation process.

Is there benefit in further optimising the way that NEAT does inter-species mating for the purposes of cross breading the intelligence gained from these trainings?

Is there a simple benefit of speeding up the running of the training by moving Mario to these points?

[Figures: Overlay of all the experiments on a fitness vs. generation line plot ]

[Some stats on how quickly it overcame 1 or more stuck areas?]

## Conclusions and Future Work

While it is too early to be making conclusions, we feel that by focusing a Neural Network on what is effectively rich, significant information rather than having it repay insignificant and general parts of the game again and again, will show that in the genetic evolution approach favours a rich information environment and quickly builds neural networks that are more valuable than ones that done.

We believe that future work could be done to improve how networks are merged.

There could be different strategies for detecting stuck states, or other important parts of a game for the purpose of replaying them.

We also believe this could be interesting to take into other games such as sonic to see how it handles there.

Formula to consider:

Given a genome that is making progress at the rate fitness/generation. Having potential gain of x over a series of future generations. If the gain of x is less that the fitness/generation rate of a new network in training on that specific area of the game, there is a potential for a beneficial boost in drilling a new network/ merging a new network rather than continuing with the current training.

## Bibliography

[1] Kenneth O. Stanley. “Efficient Evolution of Neural Networks Through Complexification” Ph.D. Thesis; Department of Computer Sciences, The University of Texas at Austin. Technical Report~AI-TR-04-314, August 2004

[2] Kenneth O. Stanley and Risto Miikkulainen. “Evolving Neural Networks Through Augmenting Topologies” Evolutionary Computation 10(2):99-127, 2002.