Combating Stagnation in Reinforcement Learning Through 'Guided Learning' With 'Taught-Response Memory'*

Keith Tunstead[1][0000−0002−9769−1009] and Joeran Beel[1][0000−0002−4537−5573]

Trinity College Dublin, School of Computer Science and Statistics, Artificial Intelligence Discipline, ADAPT Centre, Dublin, Ireland
{tunstek,beelj}@tcd.ie

Abstract. We present the concept of Guided Learning, which outlines a framework that allows a Reinforcement Learning agent to effectively 'ask for help' as it encounters stagnation. Either a human or expert agent supervisor can then optionally 'guide' the agent as to how to progress beyond the point of stagnation. This guidance is encoded in a novel way using a separately trained neural network referred to as a 'Taught Response Memory' that can be recalled when another 'similar' situation arises in the future. This paper shows how Guided Learning is algorithm independent and can be applied in any Reinforcement Learning context. Our results achieved superior performance over the agents non-guided counterpart with minimal guidance, achieving, on average, increases of 136% and 112% in the rate of progression of the champion and average genomes respectively. This is due to the fact that Guided Learning allows the agent to exploit more information and thus, the agent's need for exploration is reduced.

Keywords: Active learning · Agent teaching · Evolutionary algorithms · Interactive adaptive learning · Stagnation

1 Introduction

One of the primary problems with training any kind of modern AI in a Reinforcement Learning environment is stagnation. Stagnation occurs when the agent ceases to make progress in solving the current task prior to either the goal or the agents maximum effectiveness being reached. The reduction of stagnation is an important topic for reducing training times and increasing overall performance in cases where training times are limited.

This paper will present a method to reduce stagnation and define a framework for a kind of interactive teaching/guidance where either a human or expert agent supervisor can guide a learning agent past stagnation.

* This publication emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number 13/RC/2106.

In terms of related work, we will briefly discuss Teaching and Interactive Adaptive Learning. The concept of Teaching[3] encompasses agent-to-agent [6], agent-to-human [8] and human-to-agent teaching [1]. Guided Learning is a form of Teaching that can take advantage of both human-to-agent and agent-to-agent. Interactive Adaptive Learning is defined as a combination of Active Learning, a type of Machine Learning where the algorithm is allowed to query some information source in order to obtain the desired outputs, and Adaptive Stream Mining which concerns itself with how the algorithm should adapt when dealing with time changing data [2].

2 Guided Learning

Guided Learning encodes guidance using what we refer to as Taught Response Memories (TRMs), which we define as: a memory of a series of actions that an agent has been taught in response to specific stimuli. A TRM is an abstract concept but its representation must allow for some plasticity in order to adapt the memory over time, this allows a TRM to tend towards a more optimal solution for a single stimulus or towards its applicability, more generally, to other stimuli. In this paper we represent TRMs as separately trained feed-forward neural networks. TRMs may consist of multiple actions and this can cause non-convergence when conflicting actions are presented, therefore we define a special case TRM, referred to as a Single Action TRM (SATRM). Using SATRMs, multiple actions can be split into their single action components, therefore removing any conflicting actions. Due their independence from the underlying algorithm, TRMs (and subsequently Guided Learning) can be used with any Reinforcement Learning algorithm.

The ideal implementation of Guided Learning can be best described using an example. In the game Super Mario Bros, when a reinforcement agent stagnates at the first green pipe (see Fig. 1 in Appendix A), the agent can request guidance from a supervisor. If no guidance is received within a given time period, the algorithm will continue as normal. Any guidance received is encoded as a new TRM. The TRM can be 'recalled' in order to attempt to jump over, not only the first green pipe but the second, and the third and so on. A TRM is 'recalled' if the current stimulus falls within a certain 'similarity threshold', $\theta < t$, of the stimulus for which the TRM was trained, i.e. $\theta = \arccos \frac{a.b}{|a||b|}$ where a and b are the stimulus vectors. Because each TRM is plastic, it can tend towards getting more optimal at either jumping over that one specific green pipe or jumping over multiple green pipes. This also helps in cases where guidance is sub-optimal. A full implementation of Guided Learning can recall the TRM, not only in the first level or in other levels of the game but in other games entirely with similar mechanics to the original game (i.e. another platform or 'jump and run' based game, where the agent is presented with a barrier in front of it). For more information please refer to the extended version of this manuscript [7].

3 Methodology

The effectiveness of a limited implementation of Guided Learning[1] will be measured using the first level of the game Super Mario Bros[2]. The underlying Reinforcement Learning algorithm used was Neural Evolution of Augmenting Topologies (NEAT)[5]. NEAT was chosen firstly due to it's applicability as a Reinforcement Learning algorithm and secondly due to NEATs nature as an Evolutionary Algorithm. The original intent was to reuse TRMs across multiple genomes. While this worked to an extent (see Avg Fitness metric in Fig. 3 in Appendix B.1), it was not as successful as originally hoped. This is because different genomes tend to progress in distinct ways and future work still remains in regards to TRM reuse. Stagnation was defined as evaluating 4 generations without the champion genome making progress.

To evaluate Guided Learning, a baseline was created that only consisted of the NEAT algorithm. The stimulus was represented as raw pixel data with some dimensionality reduction (see Fig. 2 in Appendix A). The Guided Learning implementation then takes the baseline and makes the following changes: 1) Allows the agent to 'ask for help' from a human supervisor when stagnation is encountered. 2) Encodes received guidance as SATRMs. 3) Activates SATRMs as 'similar' situations are encountered.

Both the baseline and Guided Learning algorithms were evaluated 50 times, each to the 150th generation. 'Best Fitness' and 'Average Fitness' results refer to the fitness of the champion genome and average fitness of the population at each generation respectively. Where 'fitness' is defined as the distance the agent moves across the level.

4 Results & Discussion

For Guided Learning, an average of 10 interventions were given over an average period of about 8 hours. Interventions were not given at each opportunity presented and were instead lazily applied, averaging to 1 intervention for every 3 requests. The run-time of Guided Learning was mostly hindered by the overhead of checking for stimulus similarity, this resulted in an extra run-time of about 2x the baseline. This run-time can be substantially improved with some future work.

Guided Learning achieved 136% and 112% improvements in the regression slopes for both the Mean Best Fitness and Mean Average Fitness respectively (see Fig. 3 in Appendix A). We also looked at the best and worst performing cases. These results can be seen in Fig. 4 and Table 2 in Appendix B.2.

[1] https://github.com/BeelGroup/Guided-Learning

[2] Disclaimer: The ROM used during the creation of this work was created as an archival backup from a genuine NES cartridge and was NOT downloaded/distributed over the internet.

The results obtained show good promise for Guided Learnings potential as such results were obtained with only a partial implementation and much future work still remains.

Some of the limitations of Guided Learning include the need for some kind of supervisor, its current run-time and its domain dependence i.e. a TRM for 'jump and run' games would not work in other games with different mechanics or reinforcement scenarios.

Future work will include: 1) Building Guided Learning using more state of the art Reinforcement Learning algorithms [4]. 2) Using a more generalized encoding of the stimulus to allow TRMs to be re-used more readily while still balancing the false-negative and false-positive activation trade-off (i.e. feeding raw pixel data into a trained classifier). 3) Implementing TRM adaptation. 4) Taking advantage of poorly performing TRMs as a method of showing the agent what *not* to do [3]. 5) Run-time optimization by offloading the similarity check and guidance request to separate threads, this would mean that the agent would no longer wait for input and TRM selection predictions can also be made as the current stimulus converges towards a valid TRM stimulus.

References

1. Hussein, A., Elyan, E., Gaber, M.M., Jayne, C.: Deep reward shaping from demonstrations. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 510–517. IEEE (2017)

2. Kottke, D.: Interactive adaptive learning (2018), http://www.daniel.kottke.eu/2018/tutorial-interactive-adaptive-learning, [Online; accessed June 18, 2019]

3. Lin, L.J.: Self-improving reactive agents based on reinforcement learning, planning and teaching. Machine learning **8**(3-4), 293–321 (1992)

4. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)

5. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary computation **10**(2), 99–127 (2002)

6. Taylor, M.E., Carboni, N., Fachantidis, A., Vlahavas, I., Torrey, L.: Reinforcement learning agents providing advice in complex video games. Connection Science **26**(1), 45–63 (2014)

7. Tunstead, K., Beel, J.: Combating stagnation in reinforcement learning through 'guided learning' with 'taught-response memory' [extended version]. arXiv (2019)

8. Zhan, Y., Fachantidis, A., Vlahavas, I., Taylor, M.E.: Agents teaching humans in reinforcement learning tasks. In: Proceedings of the Adaptive and Learning Agents Workshop (AAMAS) (2014)

A Figures & Tables

Fig. 1. First pipe encounter in Super Mario Bros.

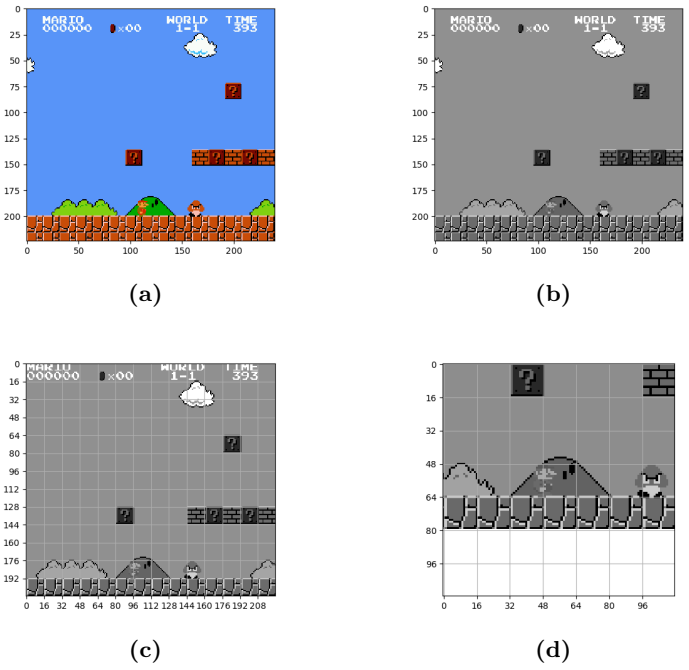


Fig. 2. Input Reduction Pipeline Examples. **(a)** Raw RGB Frame **(b)** Grayscaled Frame **(c)** Aligned and Tiled Frame **(d)** Radius Tiles Surrounding Mario, $r = 4$

Table 1. NEAT Configuration Used During Evaluation

Parameter	**Value**
Initial Population Size	50
Activation Function	Sigmoid
Activation Mutation Rate	0
Initial Weight/Bias Distribution Mean	0
Initial Weight/Bias Distribution Std. Deviation	1
Weight & Bias Max Value	30
Weight & Bias Min Value	-30
Weight Mutation Rate	0.5
Bias Mutation Rate	0.1
Node Add Probability	0.2
Node Delete Probability	0.1
Connection Add Probability	0.3
Connection Delete Probability	0.1
Initial number of Hidden Nodes	6
Max Stagnation	20
Elitism	5

B Results Figures & Tables

B.1 Average Results Over 50 Trials

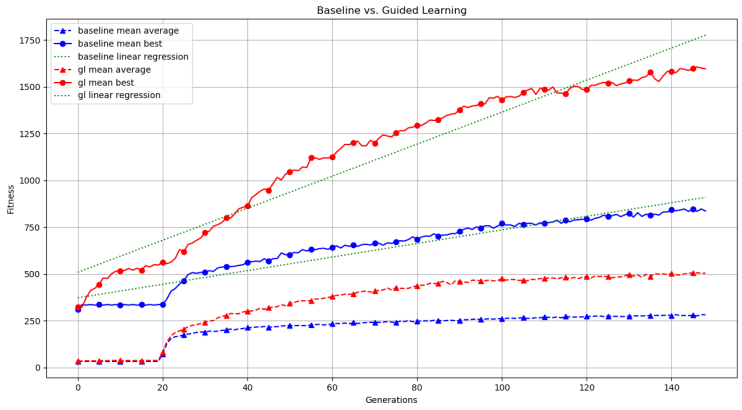
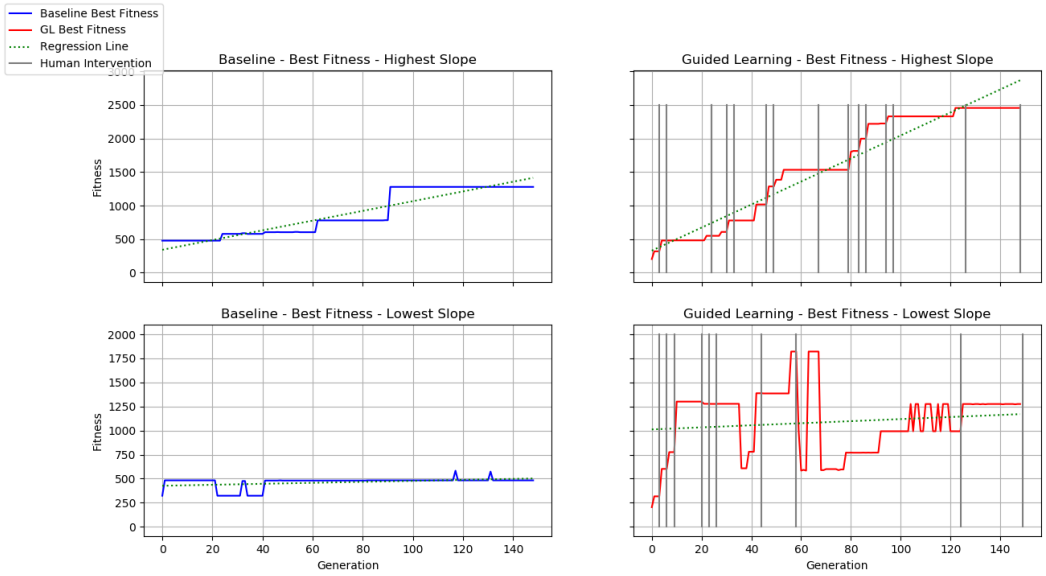
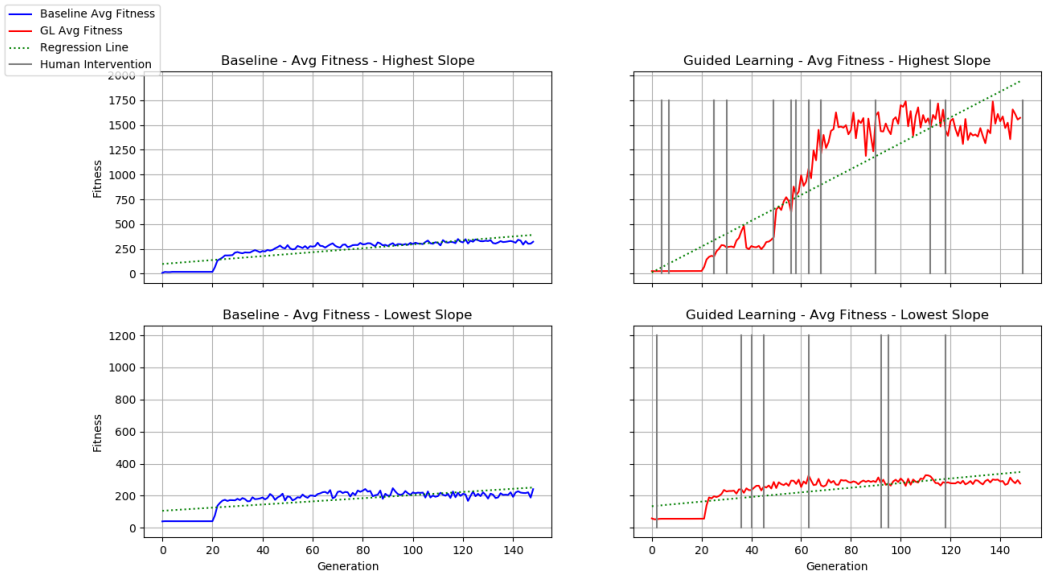


Fig. 3. Baseline vs. Guided Learning Average Results Per Generation (Higher is better).

B.2 Best & Worst Case Results



(a)



(b)

Fig. 4. Baseline vs. Guided Learning Best and Worst Case Results (Higher is better). **(a)** Best Fitness. **(b)** Avg Fitness.

Table 2. Baseline vs. Guided Learning Best and Worst Case Slope Results

	Baseline	**Guided Learning**	**Improvement**
Best Fitness (Highest Slope)	7.25	17.16	137%
Best Fitness (Lowest Slope)	0.51	1.07	110%
Avg Fitness (Highest Slope)	1.98	13.03	558%
Avg Fitness (Lowest Slope)	0.98	1.44	47%