Learning to control forest fires with ESP

Marco Wiering MARCO@CS.UU.NL
Filippo Mignogna FMIGNOGN@CS.UU.NL
Intelligent Systems Group, Institute of Information and Computing Sciences, Utrecht University

1. Introduction

Reinforcement Learning (Kaelbling et al., 1996) can be used to learn to control an agent by letting it interact with its environment. In general there are two kinds of reinforcement learning; (1) Value-function based reinforcement learning, which are based on the use of heuristic dynamic programming algorithms such as temporal difference learning (Sutton, 1988) and Q-learning (Watkins, 1989), and (2) Evolutionary algorithms such as genetic programming (Koza, 1992), Symbiotic Adaptive Neuron Evolution (SANE) (Moriarty & Miikkulainen, 1996), and Enforced Sub-Populations (ESP) (Gomez & Miikkulainen, 1998). There is still an ongoing debate which of these algorithms works best for a particular problem. E.g. for learning to play games, often value-function based RL seems appropriate since the Markov assumption holds. E.g., Tesauro (1992) used temporal difference learning to let a program learn to play backgammon by playing against itself, and this led to human-expert level. However, for non-Markovian environments evolutionary approaches may sometimes be more beneficial.

2. Forest Fire Control

In our current research we study forest fire control by a learning multi-agent system. For this we developed a forest fire simulator based on a stochastic cellular automaton where single cells may contain trees, grass, water, be on fire or not, etc. The fire starts at some place and then propagates itself according to wind strength and direction, and humidity. Also because different cells (grass and trees) have different thresholds for starting to burn and spread different rates of fire activity, the dynamics of the forest fire can be quite complex.

The goal of the multi-agent system is to control the propagation of the forest fire. This they can do by cutting firelines around the fire. Therefore the question becomes where should the agents cut firelines to minimize the damage done by the forest fire?

To study this problem, we first notice that the Markov property does not hold even in the case of a single agent. That's because the fireline should be around the complete fire. The last action of an agent is only good if the agent has already cut an almost complete fireline around the fire, and therefore the reward depends on the previous actions of the agent. In the case of multiple agents the problem even becomes harder. To solve the problem, we need to generate subgoals and then cut firelines between subgoals. The planning between subgoals is currently done using A* and this takes into account that cutting firelines over grass can be done much faster than cutting away trees. The main problem is to generate subgoals which are optimal. This is a difficult control problem, since each forest fire looks different and the state space is huge (i.e., the forest fire may consist of more than 10,000 burning cells).

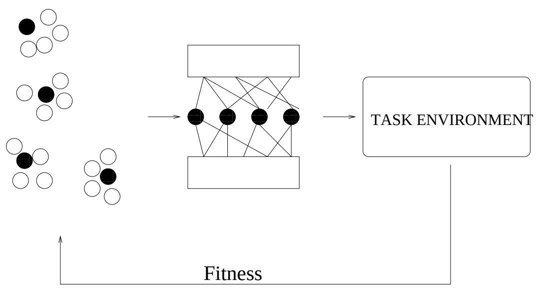

Figure 1. Enforced Sub-Populations (ESP) constructs a neural network by taking one neuron from each subpopulation. The resulting neural network is tested and the evaluation is used to evolve novel sub-populations of neurons.

3. ESP

For learning to generate subgoals, we use Enforced Sub-Populations (ESP). ESP (Gomez & Miikkulainen, 1998) is an evolutionary method for evolving neural networks. It works by keeping different subpopulations containing neurons which have weighted connections to inputs and outputs (see Figure 1). To generate a neural network, one neuron is selected from each subpopulation and these neurons then form a feedforward neural network. In order to train the system, each neuron from each subpopulation is combined a number of times with neurons from different subpopulations, and the neuron is assigned the average fitness of the networks in which it took part. Then crossover and mutation are used within the subpopulations to generate new neurons. In this way, neurons which can collaborate well with other neurons will receive higher fitness values, and will be used to evolve novel neurons. ESP has already been used for particular difficult reinforcement learning problems such as double pole balancing with hidden state (Gomez & Miikkulainen, 1998) and obtained good results.

Another advantage of ESP is that it is easy to use for multi-agent learning. In multi-agent learning, issues arise about credit assignment to individual agents given a team reward. In ESP these issues are solved using the same mechanism as with single agent learning; each agent uses its own neurons and each neuron is again evaluated by how well the resulting combinations of neurons work.

For controlling forest fires, one agent first uses a neural network to place an initial possible subgoal, and then it uses a second neural network to see whether neigboring states are better suited as subgoal. Thus the second neural network is used to refine the subgoal's location and continuously selects new neigboring states with the highest value according to the second neural network until a state is reached with the highest value. The neural networks receive relative inputs to consider the state; these inputs consist of many features such as the distance of a possible subgoal to the centre of the fire, the distance to the current state of the agent, the distance to the east/north/west/south point of the fire, etc. In case of multi-agent learning, the agents also receive information about the location and committed subgoals of other agents.

4. Preliminary Experiments

We have done several preliminary experiments to see whether the learning system is able to learn to set subgoals in a good way. A problem is that in the beginning, very few controllers learn to completely surround the forest fire, and therefore there is little selective pressure. However, if we first train the system on slowly propagating fires, the system reliably learn to set subgoals which surround the fire. After this, the fire can propagate faster, and the neural networks are refined to deal with the more complex problems. The learning time takes about half an hour in which about 10,000 controllers are evaluated. When we look at the determined subgoals using our simulator, they make a lot of sense. They are not too close to the fire, but neither too far away.

5. Conclusion

We are studying methods for controlling forest fires with multiple agents. For this we have implemented a forest fire simulator with which we will compare a number of multi-agent learning methods. Currently we have only implemented ESP, an evolutionary method for evolving neural network controllers. Current experiments have shown that ESP is able to learn to generate subgoals for a single agent, so that the fire propagation is stopped within a reasonable amount of steps. In the near future we will run much more experiments and also study cooperative learning with multiple agents.

References

Gomez, F., & Miikkulainen, R. (1998). 2-d pole balancing with recurrent evolutionary networks. *International Conference on Artificial Neural Networks* (pp. 425–430). Berlin, New York: Springer.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Koza, J. R. (1992). Genetic evolution and co-evolution of computer programs. In C. Langton, C. Taylor, J. D. Farmer and S. Rasmussen (Eds.), *Artificial life ii*, 313–324. Addison Wesley Publishing Company.

Moriarty, D. E., & Miikkulainen, R. (1996). Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, *22*, 11–32.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Tesauro, G. (1992). Practical issues in temporal difference learning. *Advances in Neural Information Processing Systems 4* (pp. 259–266). San Mateo, CA: Morgan Kaufmann.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. Doctoral dissertation, King's College, Cambridge, England.