Chris Bush

NSC 309

Project Proposal

Fall 2007

Currently in Nero, agents learn through a reward based "trial and error" system on how to overcome obstacles in the game. For instance, if an agent is instructed (e.g. rewarded) for approaching a flag, but there is a wall in the way, the agents that randomly make it around the wall will be the ones to pass on their genes to the next generation. Over time, the agents will learn to make it around the wall and approach the flag. The same applies for any other activities in the game. While this is a successful, basic model of machine learning, it does not simulate many of the real-life roles educated people take on to teach subordinates to tackle problems.

The question that needs to be answered is can agents not only learn to accomplish tasks, but can they also learn to be leaders to lead and teach other agents to accomplish those tasks? Following the military model in Nero, the basic question I want to answer is what will happen if, after an initial training period, agents are assigned "ranks" and are allowed to advise other agents on actions to take? These ranks will be determined either by the user manually assigning ranks to agents, or the ranks will be determined automatically through agents being "promoted" during a "promotion mode" of the game. An agent that is constantly promoted and used for evolution will be promoted above agents that do not do as well. To avoid difficult "communication" research problems, there should also be new sliders that agents will react to. These sliders will be things such as "follow the closest higher ranked leader", "protect the higher ranked leader" by always staying in front of the leader, among others.

This will allow many interesting AI questions to be looked into and possibly answered. For instance, once agents are ranked, who will be promoted faster – agents that follow their leaders, or "top gun" units that don't listen to their officers? Will squads that have a strict hierarchy they follow perform better in combat? Will agents learn faster by mimicking what their leaders are doing? Will it actually hinder evolution and learning because the difference between leader and subordinate will become small? Can the leader take advice from subordinates when they stumble across the solution to an obstacle that is better than what the leader has come up with? What happens if a ranked agent is placed in a brand new population, and that population is set to mimic the leader completely? What if that same population is set to NOT do anything the leader does? Will they eventually learn new ways to accomplish the same tasks, and will new, possibly smarter leaders emerge?

An initial experiment to begin tackling this problem would be to simply allow the user to click on an agent and assign a rank. There will be three ranks – private, sergeant, and captain. Privates listen to sergeants and captains, and sergeants listen to captains. There can only be one captain at a time, and there can only be at most one sergeant for every three privates. Two sliders will then be initially implemented. First, one will be implemented for reward for sticking together with the next ranked leader. This way privates won't listen to captains, making for a more interesting experiment as "advice" is sent down the chain of command. Second, if there is time, one will be implemented for protecting the next ranked leader (privates will generally stand in front of sergeants, and sergeants in front of captains, but if mimic is set as well, it will be hard to mimic while staying in front at all times). There will also be an option (a button, most likely) to turn off rank at any point during the game, to see if agents can act on their own accord without their officers present, or if everything goes chaotic.

Currently, there is work being done in the human-agent training and interaction area to determine how agents react to a human guiding and training them to complete these tasks (2) (4). This research problem essentially takes this research a step further to determine if the agents can then teach their subordinates the tasks they have learned and been taught. The long-term idea and potiential goal is to expand upon the rtNEAT model, where agents learn in real time as the game is played, to allow those units to leverage rtNEAT to learn to teach each other in the game (1). Certain elements of rtNEAT will need to be reevaluated to allow for the research – for instance, instead of just

removing the agents with the lowest fitness (3), agents will be assigned to certain ranks, and the lowest among those ranks will be removed. That way, everyone will not converge towards the captain rank described above. All agents will constantly move up within their ranks, so the ranks themselves will improve. Some interesting related work done in this area was by Chern Han Yong et al. to allow for advice to be sent to and between units given a specified grammar. For instance, "through exploration, they will eventually learn to head away from the flag and go around the wall to get to their target efficiently. Although such learning does not take very long in terms of game time — about one minute on average — it still seems like such a simple solution is not worth the time to be discovered automatically, since it is obvious to the human player what the agents should do. Such knowledge can be easily encoded as advice, as shown in Figure 3. It tells the agent how to go around the right side of a wall that is initially in front of it, by turning right and moving along the wall. "(4) This research proposal simply takes the idea of providing advice from human to agents to the agents themselves. The agents giving the advice will be determined by better agents (which is determined by rtNEAT), and the agents receiving the advice will be the newer and "dumber" agents. This will allow the agents to learn whether or not they want to take advice from other agents, as well as hopefully learn successful methods and ways to use that advice to succeed in their tasks.

Bibliography

1.) Stanley, Kenneth et al. "Real-Time Learning in the NERO Video Game"
http://nn.cs.utexas.edu/downloads/papers/stanley.aiide05demo.pdf

2.) Craver, James. "Human Examples in Evolution"

http://www.cs.utexas.edu/~risto/nsc309/private/slides/jcraver.pdf

3.) Stanley, Kenneth et al. "Evolving Neural Network Agents in the NERO Video Game"

http://nn.cs.utexas.edu/downloads/papers/stanley.cig05.pdf

4.) Yong, Chern Han et al. "Incorporating Advice into Evolution of Neural Networks"

http://nn.cs.utexas.edu/downloads/papers/yong.gecco05lbp.pdf