Curiosity as a Survival Technique

Amber Viescas
Department of Computer Science
Swarthmore College
Swarthmore, PA 19081
aviesca1@cs.swarthmore.edu

Anne-Marie Frassica
Department of Computer Science
Swarthmore College
Swarthmore, PA 19081
afrassi1@gmail.com

May 12, 2009

Abstract

In this paper we present a system that motivates adaptive, curious behavior in a robot in a survival scenario. We continue prior work done on using evolutionary algorithms and neural nets to develop a robot controller capable of collecting energy in a simulated survival environment. Neural nets are efficient for producing survival behavior, but the robot only knows how to respond to stimuli it has seen before - it cannot handle situations it has not trained for. In this study, we couple Intelligent Adaptive Curiosity [1] with a NEAT-evolved [2] survival brain to encourage exploration and learning new situations. We find that this "dual-brain" approach encourages intelligent exploration and learning in the robot, which benefits survival, and that the survival brain gives the curious brain time to do this learning. Our system shows a statistically significant increase in survival time as compared to two baseline systems, a pure survival brain and a survival brain augmented by random movements.

1 Introduction

In this experiment, we approach the topic of adapative learning in a developmental framework. In previous experiments, we had used NEAT to evolve a general"survival" brain within an environment with "food" objects and "hazardous" objects, which we saw produced effective food-gathering and hazard-avoidance behavior but could not handle complex environments. Despite being evolved with a general fitness function (number of steps survived), the robot's behavior was simplistically task-oriented, and in a scenario in which food was not immediately available, the robot had difficulty adapting. We now seek to produce self-motivated behavior and to evaluate its contribution to the success of a robot in an environment with rewards, hazards, and learning scenarios. The current experiment supplements the survival brain with an artifical curiosity mechanism and tests the resulting "dual-brain" in a variable environment where more food can be discovered through curiosity. We adapt a system of Intelligent Adaptive Curiosity [1] (IAC) to our task in order to produce this behavior, and we find that curiosity does not always kill the cat - in fact it is beneficial in the survival scenario. Our test environment simulates a world with food for energy, hazards that randomly hurt or help the robot, and objects that produce predictable and random effects. Robots are evaluated on their ability to survive by collecting energy.

In the Related Work section we review briefly the literature on the evolution of neural networks, as well as fitness functions and design choices, and we discuss the IAC approach to developmental learning. In the Experiments section, we describe the environment and "dual-brain" system in detail, our hypotheses, and the baseline tests to which we compare our outcomes. In the Results section we provide a qualitative and quantitative discussion of these experiments, and lastly, we discuss the relevance of our results to the literature.

2 Related Work

The present work seeks to produce food-gathering and danger-avoidance behavior in a simulated robot through both an evolved brain and IAC. We capitalize on previous work for the structure of our approach - in the overall definition of the fitness of each evolved robot-controller, the method used to evolve them, and the implementation of curiosity.

Previous efforts have shown value of letting a robot design its own solution to a task. Floreano and Mondada [3] worked on evolving a Khepera robot to locate and navigate to a battery charger. They discover that allowing a fitness function to develop naturally produces better behavior than trying to define a specific function for the task. Their solutions show that pre-determining a complex fitness function is not helpful to the evolutionary process, which, biologically, is in fact based on the most general fitness function of all - survival.

In prior research, we used an implementation of NEAT to produce successful food-gathering and hazard-avoidance behavior in our robot. The NEAT algorithm [2] describes a complexifying approach to evolution. The topology of a neural network is allowed to mutate incrementally in hopes of building a search space that is appropriate for the task, rather than guessing at this design a priori. For the simple task of survival in the environment shown in Figure 1, where green lights allow the robot to survive longer, red lights hurt survival chances, and fitness is defined as the length of time survived, about 20 generations of NEAT could produce a simple neural network to solve the task, such as Figure 2, or a very complex one (with no benefit to behavior), in Figure 3.

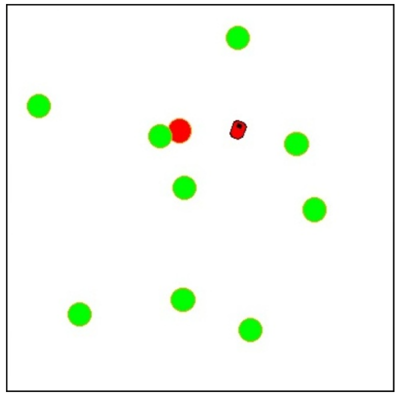

Figure 1: Simple Survival Environment

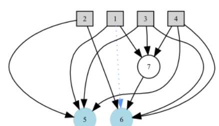

Figure 2: Simple Evolved Network

In evaluating the efficacy of a NEAT-evolved brain for this task, we found the quantitative results encouraging; the robots survived longer with each generation and eventually all generally managed to find all the food and avoid an early death. But in the end, their behavior was boring - if the robot collected all the food in the environment, it merely sat in place. In this work we show how a method of implementing curiosity, IAC [2], can be used to produce self-motivated exploratory behavior. IAC was developed by (Oudeyer et. al. 2007) as a method of implementing curiousity as a mode of development. The authors describe a system that separates the sensorimotor contexts (roughly, experiences of the robot) and tracks the robot's progressive ability to predict the result of moving within those regions. The system tries to keep the robot moving towards regions with high learning progress, i.e., where the robot is reducing its error in predicting sensor values based on motor actions in that region. The logical structure of the system is depicted in Figure 4.

At each time step, the new sensor values are input to the brain, which compares them to its prediction from the last step. It updates the error for the pertinent region and stores a new exemplar there as well. The learner for that region, called the expert, trains over this data continuously and is later called upon to make predictions for future sensor values and to choose a motor

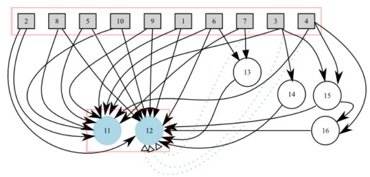

Figure 3: Complex Evolved Network

action. Then, the current sensor context is compared to past contexts seen, and the appropriate region is identified. The expert in that region, as mentioned above, both chooses the next motor outputs and makes a prediction based on the output chosen.

Building on this related work, we evaluate a “two-brain” system. The “survival” brain is developed through the NEAT process and does not evolve throughout our current experiment. The “curious” brain learns throughout the experiments we describe below. At each time step, our system decides which brain’s output to send to the robot’s motors. We describe this process in detail in the next section.

3 Experimental Setup

To test the effect of curiosity on survival, we allowed both the survival brain and the curious brain to control the robot per the method described below. We also set up two baseline experiments to serve as a benchmark for comparison. In all of experiments, the robots are evaluated on the number of time steps they survive.

3.1 Environment and Sensors

All experiments are run in a simulated environment within Pyrobot [4]. The environment is an open arena (no walls) with a configuration of lights in the center. The robot’s start position is in the center of the lights, facing north, and the robot begins each trial with 100 energy points, which will allow it to survive initially for 100 time steps. There are three types of lights, and four instances of each type. Green lights signify food, or energy, and always give the robot 50 energy points and disappear on doing so. Red lights are hazards, as they randomly affect the robot’s energy, increasing or decrementing it by some value within ten points. Blue lights represent transportation to a new area; the current implementation repositions the robot to its initial coordinates and regenerates the green lights without directly affecting the robot’s energy level. The lights are always initialized in the same position. The result of these initial experiments is the environment in Figure 5.

In order to survive in the environment, the robot-controller requires infor-

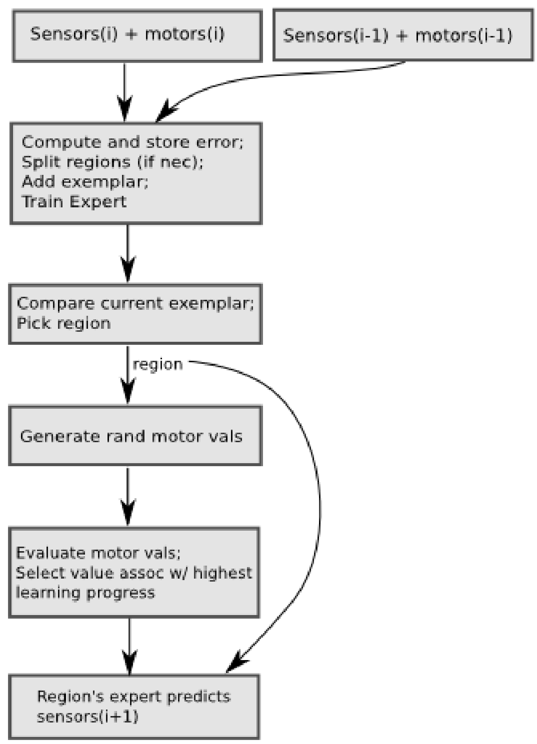

Figure 4: Flow Chart for IAC at Timestep i

mation on the lights surrounding it, and produces values for the left and right motors. The values are scaled to values between -1 and 1, with 0 representing no motion in the motor. The simulated robot has two light sensors, no sonar information and no camera. The robot is only equipped with light sensors on its front side. Each light sensor gives two readings: the brightness of the light and its RGB value, and we preprocess the light input so that the robot can detect amounts of green, red, and blue light, which are the values given as input to both survival and IAC brain. This preprocessing step also represents a reduction in complexity in order to facilitate the experiment - red, green, and blue lights do not overlap in RGB value, thus simplifying our task. In addition, we combine the overall brightness values with the proportion of colored light to get a single value–the amount of mono-colored light it observes. Also, the IAC brain requires as input the amount of energy the robot has at each time step. Energy is not detected through a sensor but rather is given to the curious brain

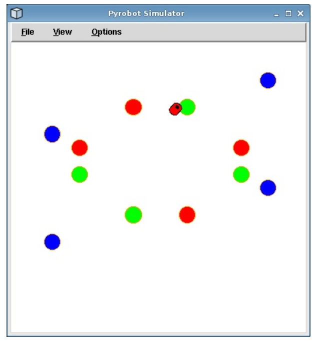

Figure 5: Resulting Environment

directly. Although this represents some human intervention, it is not an unreasonable sensor for a creature to have (whereas, its global position, for instance, might be).

3.2 Survival Brain

As we mentioned above, at each step the motor output comes from one of two brains. The survival brain is a neural net evolved through NEAT. Its weight and topology remain fixed throughout the current experiment. The neural net was evolved by evaluating individuals in the environment shown in Figure 5. The phenotype of each individual controller was generated randomly, through the evolutionary process, and then allowed to run for three trials in the survival world. In this evolution, the environment contained only green lights, potential energy. The robot was again given only light sensors, and the green value and brightness measure was given as input to the neural net. At each time step, the robot lost an energy point, and could only increase its lifespan by collecting green lights. The fitness function used to produce this behavior was general: the number of time steps survived. In about 20 generations, NEAT found solutions to this problem, namely, robots that moved towards green light when they detected it. In the current experiment we use the topology in Figure 2 to produce motor output. The survival brain generates a motor output suggestion at every time step although it may not always be executed.

3.3 Curious Brain

The second brain used to generate motor outputs is the curious brain, which is an implementation of IAC. This brain is not fixed at the start but rather constantly learns as it sees more input. The curious brain takes light values and the robot's energy value as input. Even when the curious brain's motor outputs were not executed, it learns about the robot's environment. So, at each time step, the curious brain evaluates its sensorimotor context (based on current sensors and its previous motor actions). It does two things: computes the prediction error from the last time step, and makes a new motor output suggestion and prediction. If the last motor output was suggested by the survival brain, it makes a prediction about that and compares it to the current sensor context, just as it would have if the motor decision was generated by the curious brain itself. After computing and updating the error for a region, the curious brain compares its sensorimotor context to exemplars of regions it has seen before, picks a motor output (out of a randomly generated set) that will place it nearest the region with the highest learning progress, and then uses the expert in the relevant region to predict its next sensor values (light and energy values).

We expect that as a trial progresses, the green lights will become “boring” (have low learning progress), because they are very predictable, that red lights will also become “boring”, because they are completely unpredictable, and that blue lights will be interesting for a while, because interaction with them will be rare (as soon as the robot moves over them, it is moved back to its starting position), but they are predictable. Finding the blue lights should lead the robot to more food for at least as long as the blue lights remain interesting. In fact, our trials do not last long enough to show discrete learning stages, and we discuss the reasons for that in Results. To test our implementation of IAC, we allowed a robot to survive for thousands of steps in a slightly altered environment. For this test, there was one red, one blue, and one green light. The green light did not disappear when the robot passed over it, instead remaining as a continual source of food, and the blue light simply repositioned the robot. The red light was as described above. We found that in general, the results of the robot’s focus are as shown in Figure 6. The x-axis represents time, and the y-axis, percentage of time spent on top of a particular color of light. The red and green lines show alternating peaks that diminish in height over time; the robot is first interested in one, and then the other, and then eventually in neither as it learns to predict them or stop trying. The blue line always seems low, in fact it should be because the robot cannot focus on blue for very long, but the peaks in blue increase as red and green decrease. This is essentially the behavior we predicted.

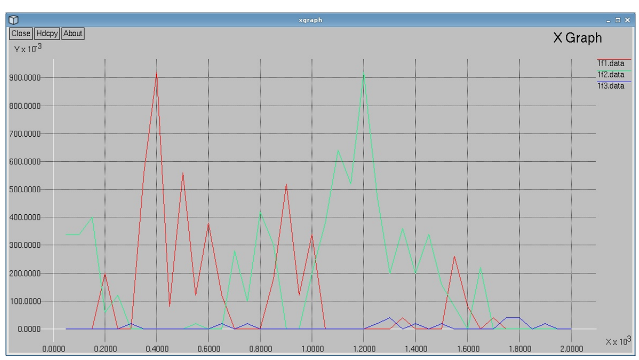

Figure 6: Focus of Curious Brain over 2000 Steps

3.4 Decision Tree

At each time step both survival and curious brain generate a suggested motor output. Survival brain always generates the motor action that will place the robot nearest a green light. Curious brain generates a motor action that will push the robot towards an interesting region (one with high learning progress). The execution of one of these outputs is decided by a decision tree, shown in Figure 7. The idea is this: if the robot is running low on energy (or "hungry") and there are lights available and in sight, it ought to move towards them; if the robot is low on energy and there is no light available, or not in sight, it ought to explore its environment in hopes of finding more; if the robot is not low on energy it ought to learn about its environment. In this step, we do give the robot information about its environment that it did not directly collect through its sensors - the amount of lights left in the arena. This ensures that the robot does not waste time seeking lights that do not exist, but in a completely robust system, this decision might be left to the robot.

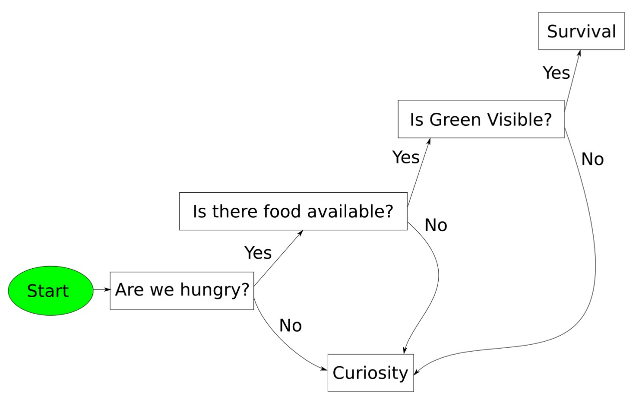

Figure 7: Decision Tree for Choosing Motor Values

The robot is initialized with a "hungry" state; this is so that survival action will prime the curiosity system so that it creates exemplars even while the curiosity system is not running.

3.5 Baseline Controllers

In order to evaluate our "dual-brain" system, we construct two baseline controllers to compare it to. These controllers are evaluated in the same environment as the survival/IAC brain. The first baseline is the simplest: just the survival brain. With this baseline, we are evaluating how long a robot can survive in this environment if it is trained only to navigate to green lights. We expect that this brain will be very efficient at finding the green lights, giving it at least 200 more energy points (4 green lights at 50 points each) in addition to the 100 starting energy points. It is possible that in the process of navigating to a green light, the robot may pass over a red light, changing its energy randomly, however the blue lights should be out of reach to this brain, and therefore no great leaps in survival time should be achieved.

The second baseline more closely mirrors the current "dual-brain" design. However, instead of switching to a curiosity mode when it has high energy or no greens are available, it choose random motor values. We expect that this brain will not only collect at least the four initial greens (through its survival brain), it will occasionally traverse a blue light, either by navigating directly to it, or by beginning its survival path to green lights from a new position, such that a blue light is between it and the nearest green. This brain may not be motivated to find blue or red lights, indeed it is not learning, but by moving randomly may encounter them.

4 Results

Over the course of our series of experiments, we achieved significantly different results between the baseline survival-only brain, the Survival/Random brain,and experimental hybrid Survival/IAC brain. Much of the qualitative observational differences we noticed corresponded to differences in quantitative performance. We performed one hundred trials on each brain and recorded the overall performance of the brains in Figure 8.

	Survival Only	Survival + Random	Survival + IAC
Average Fitness	298	295	340
Min Fitness	160	63	90
Max Fitness	422	952	1096

Figure 8: Decision Tree for Choosing Motor Values

4.1 Survival-Only

After training a controller with NEAT in an environment with only green lights, we tested the product of the evolution in our more complex environment. Having predicted that this robot would seek green lights efficiently and little else, we saw our predictions realized as the robot maneuvered to all four green lights, before stopping and doing nothing else until it died. Over one hundred trials, the robot had an average fitness of 298, and a small range (254) over the data. We suspect most of this range was due to the random effect of the red lights which were placed in an area that was in the path that the green-finding survival brain took to collect energy. In earlier experiments we noticed that it sometimes ran over blue lights entirely by accident; we reconfigured the environment so that this would not happen.

4.2 Random and Survival

Our second baseline was a hybrid brain of the NEAT-evolved survival brain and a brain which would simply pick random values for its motor values. Like the IAC hybrid, it was designed to switch to a random mode whenever its energy reached a certain ceiling value, or when there was no green available.

We observed that quantitatively, it did no better than Survival only, with an average fitness of 295. However, the range was much wider, 889. In some situations it did much worse than the Survival-only brain, usually as a result of its random algorithm taking it too far from the lit area to be able to find the green lights when it grew low on energy again. Often the random motion did not result in net movement; instead the robot would move back and forth until it grew hungry, and as such essentially wasted time while it gradually became hungry, whereas the survival-only brain would instantly seek new lights in these cases. Sometimes, however, it would find blue lights to refresh the array of greens, and therefore would last over twice as long as it would with only the original setup.

4.3 Our IAC Hybrid

Finally, we tested our NEAT-evolved controller in conjunction with the IAC brain and observed the results. We noticed an average fitness of 340 and a range of 1006. More importantly however, was the statistical significance (t-test) comparisons we made. We observed $P = 0.011$ between Survival and IAC, and $P = 0.019$ between the Random hybrid and IAC. This suggests that the performance of the two systems was dramatically different.

The qualitative observations we made seem to support this. Overall, the robot remained fixated on input (lights) and so did not wander off the open arena. In addition, the processes it used to shift itself frequently resulted in novel situations so that when the survival brain kicked in, it found new lights that it would not otherwise run over. Occasionally it would find blue lights by accident, running them over from behind while trying to predict its light sensors as they were pointed at other lights. Other times it would run over them intentionally, getting close up.

One other phenomenon we observed was a process of red fixation. As IAC tried to predict its subsequent energy values, the unpredictible change associated with red lights resulted in much initial interest. Sometimes the robot would become “bored” and wander away, and sometimes its energy would be depleted by the red before the robot could become bored. Recall that we saw this patterning in Figure 6, where the robot would fixate on one color, become bored, and place itself on another color. Gradually its periods of fixation on both green and red lights would become shorter.

5 Conclusion

We have shown that combining a NEAT-evolved brain trained in a simple environment with a curious brain that predicts its sensors can achieve greater evolutionary success in novel environments than a brain evolved with the simple fitness function alone, or a brain that generates random motions. This requires careful managing of the input arrangement and construction of the environment.

Future work in this topic may include stretching evolution over the experimental setup once again; for instance evolution may guide the robot's choice between food-seeking behavior and curious behavior, instead of having the human decide. Evolution could also learn to "fine-tune" curiousity processes, resulting in a robot choosing (based on sensor inputs) which curiosity regions are good to focus on, which would hopefully lead to a decreasing of interest in randomness, or possibly distinguish "good" randomness (for instance, lights that always add variable amounts of energy) from "bad" randomness (lights that drain variable amounts of energy). This might also be achieved through a system of internal rewards, whereby the robot is rewarded for having energy. A system of rewards must be careful not to degenerate into a task-based architecture.

In addition, we performed all our work in an open obstacle-less simulated environment. In bringing our observations to the real world, it may be necessary to add sonar/laser sensors. However, previous experiments with IAC suggest that it has difficulty processing excessive numbers of inputs. Therefore, it may be necessary to add another learning algorithm, such as Growing Neural Gas, to process the inputs before they are given to IAC.

Finally, varying the environment could result in novel situations. Actually having the blue light change the arrangement of the lights in the next configuration (representing truly novel surroundings) and combining this with a more intelligent curiosity seeker (one that distinguishes different regions of curiosity) may result in a robot seeking novel environments through blue lights–in other words, learning to find optimum curiousity points in a general fashion, instead of having to rely on pure causal relationships between motor/sensor values at one time step and the next. Unfortunately, this meta-curiosity is outside the scope of our current project.

6 Acknowledgments

We would like to thank Lisa Meeden for her guidance in this work, and our Adaptive Robotics classmates for their suggestions throughout the process.

7 References

[1] Oudeyer, Pierre-Yves, Frederic Kaplan, and Verena Hafner. (2007). Intrinsic motivation systems for autonomous mental development. IEEE Transactions on Evolutionary Computation, 2.2

[2] Stanley,K and Miikkulainen, R. (2004) Competitive Coevolution through Evolutionary Complexification. *Journal of Artificial Intelligence Research.*

[3] Floreano, D. and F. Mondada. (1996) Evolution of homing navigation in a real mobile robot. IEEE Transactions of Systems, Man, and Cybernetics-Part B: Cybernetics, 26(3).

[4] D. Blank, D. Kumar, L. Meeden and H. Yanco (2006). The Pyro toolkit for AI and robotics. AI Magazine, Vol. 27, No. 1 (pp.3950).