# Evolutionary Swarms Using Neural Networks

*Srinimisha Morkonda*
Dept. of Computer Science
University Of Colorado
Boulder, CO, USA
srinimisha@colorado.edu

*Dave Coleman*
Dept. of Computer Science
University Of Colorado
Boulder, CO, USA
david.t.coleman@colorado.edu

*Abstract*— **This paper presents an approach for developing models of swarms using an evolutionary swarm simulator with neural network-driven swarm agents. The open source ScriptBots simulator is used to rapidly evolve the agents' neural networks to optimize survivability in a mixed environment of predators and prey. The goal of the experiments is to encourage the agents to self-learn the evolutionary advantage of swarming to optimize survival and reproduction. Furthermore, the objective is to demonstrate the emergence of higher intelligence. Various parameters are tweaked and functionality modified to the simulator to achieve this goal. Final experimental results indicate that basic swarm models can be developed through this evolutionary approach.**

***Keywords- swarm; evolution; nueral networks; learning;***

## I. INTRODUCTION

The primary goal of all living organisms is the optimization of its survival and reproduction. This optimization is accomplished over their lifetime and over generations. Lifetime optimization is accomplished through experience and learning. Generational optimization is accomplished through natural selection and genetic mutations. The key to this optimization processes is the environment the organisms are living in and the survival skills they develop.

Higher-level intelligence such as found in humans demonstrates complex mental capabilities such as self-awareness, planning, and imagination that give an evolutionary advantage in survival situations [1]. In contrast, primitive single cell organisms lack this degree of intelligence. With the varied intelligence and capabilities of the organisms in the world, the concept of predatory and prey has evolved. The higher capable organisms strive on the lesser capable organisms. Predators had been grazing on microorganisms since at least 1,000 million years ago [2]. Slime mold is an example of a single cell organism that has evolved simple techniques to increase their survival. When faced with starvation, single-cell slime mold has been observed to aggregating into multi cellular organisms when there is lack of food in the environment [3].

This project is focused on addressing these objectives with primitive agents, who aim to survive and hence reproduce, in the simulated environment. The project is based off an evolutionary artificial life simulator. Results of the project and experiment show the various techniques that can be incorporated in the agents and in the environment to improve their intelligence, and hence, establish emergent behavior.

## II. SIMULATOR ENVIRONMENT

The open source ScriptBots project, developed by Andrej Karpathy [4], was chosen as the simulator for is openness, simplicity and supporting community. ScriptBots is an evolutionary artificial life simulator that can model the development and interaction of simple single-cell agents, or bots, in a petri dish-like environment that can be compared to a "primordial soup." The simulator is built on top of the popular OpenGL graphics library [5] and can run on both Unix and Windows systems.

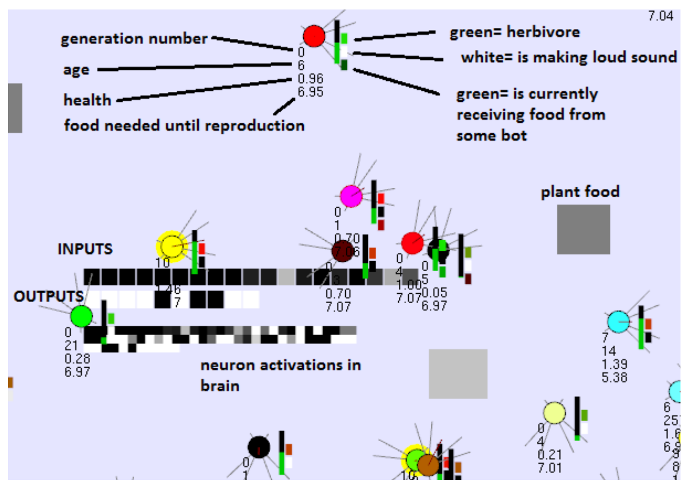


Figure 1. Diagram of bot with indicators labeled and neural network visualization.

### A. Locomotion

All agents are modeled with left and right "wheels" that can be activated at various speeds for forward movement and directional control. In addition agents have a high-energy speed boost that can allow them to temporarily move at a significantly higher speed, but at a high cost to their health.

University of Colorado Boulder, Department of Computer Science

### B. *Herbivory and Detritivory*

Agents can specialize as a carnivore, herbivore or any gradient between as an omnivore. “Plant” food exists within the world at various densities to feed the herbivores using either one of two models:

1. Food self-generates at random locations within the world using a predefined generation frequency. This frequency directly correlates to the total population in the system (carrying capacity) and the simulation run-time speed.
2. Food grows outwards in all directions from random initial locations. If agents over-consume the plant food than areas of the world can become depleted and barren. Conversely a dip in herbivore population results in a boom of plant growth at high density.

Additionally, dead agents who die without any carnivores nearby can turn into “plant” food as well, which can be thought of as decaying matter.

### C. *Predation*

Agents with a degree of carnivorous tendencies can evolve the ability to eat other agents in the simulation using “spike” weapons that protrude the front of their bodies. The agents can extend and retract their spikes at will, with a certain delay, in order to avoid spiking agents of the same species, off-spring or other evolved patterns. Using their extended spike an agent can run into another agent at a direct forward angle and damage or kill the opponent.

## III. DESCRIPTION OF AGENT INTELLIGENCE

The SACC (Sensors, Actuation, Computation, and Communication) architecture is used here to explain the intelligence and capabilities of the agents.

### A. *Sensors*

The bots has 42 sensory inputs. Three eyes (2 in front 1 in back) which have red, green, blue, and proximity sensor in every eye. Each bot has the ability to sense how much food is at its location. There is a sound sensor in each bot that responds to amount of other agents around an agent. The bots are also incorporated with a smell sensor that responds to amount of movement around it. There is a health level associated with every agent indicating how close to death an agent is. Bots can judge the health of other nearby bots using blood sensors. The hearing sensor enables agents to listen to the other agents’ shouting. They are fitted with two clock sensors that fluctuate in activity over time in different frequencies. The environment has varied temperature from one end to another. The bots have a preferred temperature randomly set in them, which decides their comfort zone. There is an input variable that can sense the outside temperature. For every tick when the bot touches the wall of the world, a sensor notifies the bot that it is too close to the wall. Figure 2 shows a model of a single bot.

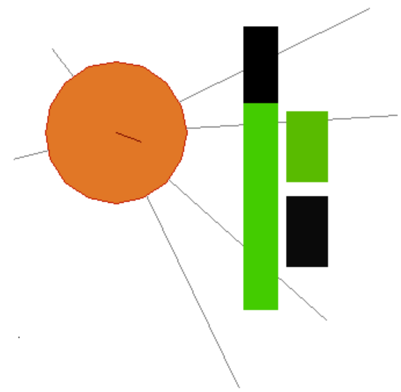

Figure 2. An image of a single agent/bot.

To introduce some randomness into the bot’s input, a random input variable is added. The bots have a positive feedback loop where the output of its current state is fed back as set of inputs. It is fed back its previous plan of the actuators and the actual output of the previous tick. By this the bots could possibly notice how different their plan was from what they actual executed.

### B. *Actuators*

The bots have 18 actuators. Two actuators determine the speed of the two wheels of the agent assisting in its movement. Three actuators determine on the amount of colors to emit where the colors are red, green and blue, just like how they can sense these colors. Carnivores have the ability to spike by which they can kill the herbivores. The longer they can spike, the quicker they can reach to the herbivore and kill. An actuator in the bot determines the length of the spikes the bot can trigger. There is a boost in the speed at which the bot can travel; this comes along with a cost, which is the loss of health. The bots have a sound multiplier that lets them be very noisy or silent. Emitting noise proportionally affects their health. At situations when an altruism actuator is activated, the bot indicates that it could share its health with others. There is another set of actuators as mentioned about which represents the next plan of the bot.

### C. *Computation*

Computation is accomplished in each agent’s brain through a recurrent neural network that is implemented as a multilayer perceptron (MLP). The traditional MLP is a network of three layers: input layer, hidden layer and output layer as shown in Figure 3.

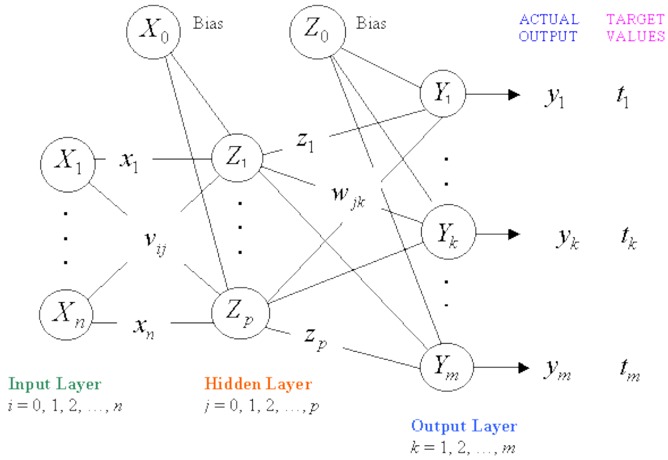


Figure 3. Traditional MLP[6].

An array of sensory inputs is presented to the input layer. Before any neurons are processed the input layer standardizes these inputs such that the range of each variable is -1 to 1. In the hidden layer, these values are randomly allocated to all the neurons that are connected from the input layer to the hidden layer. This value is multiplied by a weighted sum along which a constant value called the bias is added. The output of the hidden layer is this value after being processed through an activation function, which normally is a sigmoid function as shown in Figure 4.

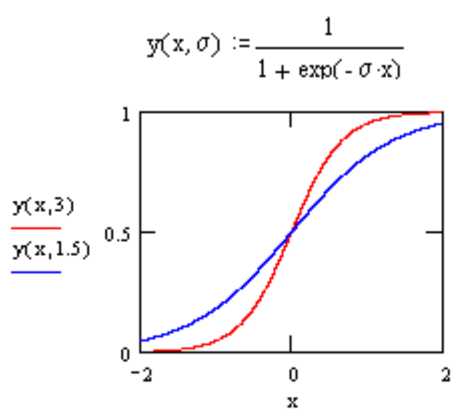


Figure 4. An example of a sigmoid function and the equation used[7].

The value from each of these neurons from the hidden layer is again multiplied by a weight. These weighted values are then combined and processed through an activation function whose output is directed as the output of the network in the output layer.

The brain size of the agents is varied throughout the experimentation from 100 to 300. The input size is 42 and output size is 18. The connections among the neurons in the network are also varied from 4 to 6. The connections represent how many other neurons each of them are connected to.

There are several differences between traditional multilayer perceptron neural networks and the MLP NN implemented in this simulator. Typically no connections are made among neurons within the same layer, whereas in this implementation, the neurons could be potentially connected to other neurons within same layer. For instance, an input neuron could be connected to another input neuron itself. Usually there neurons in the input and output layer are never directly connected while these bots could have neurons that have an input neuron connected directly to a respective output neuron. The mapping of connections is incomplete, which means not all the inputs are connected to a neuron in the brain. With the variation in the number of input and output units, not all the inputs map to the outputs judgmentally. The number of hidden units per layer is chosen arbitrarily, constrained only to computation concerns. They could be chosen to be less than the input or output units. The weights of the connections in this network are not updated until the agents reproduce.

Bots who accumulate sufficient health over time will reproduce. The simulation includes both sexual and asexual reproduction but only asexual reproduction is studied in these experiments. When a bot has reached the appropriate threshold the bot reproduces, resulting in two child bots. The child bots have a probability of genetic mutations in their brains, with slightly updated connections and weights between those connections. Due to this architecture of changing weights only when the bots enter into a new generation, learning is limited to happen only when they reproduce.

The bots have fairly a complex brain structure involving a feedback loop and the ability to plan. This is achieved by encouraging the agents to exhibit more intelligent behavior in the absence of sensory input and also to enable the bots to plan one “tick” or iteration ahead. The brains are also introduced with a stress factor to the herbivores in an attempt to make them understand the importance of moving away from carnivores to protect them just like in real life where humans could get stressed when they are attacked by wild animals. The agents also understand the increase in stress that is imposed when they are alone, hence encouraging them to stay in groups.

*D. Communication*

The bots exhibit a number of communication skills both among themselves and with the environment. A basic communication with the environment is involved in consumption of food by the bots. The bots emit colors that can be sensed by other bots. The carnivores’ spiking ability lets them to exhibit their nature of killing herbivores hence, communicate to the herbivores the same. This allows the carnivores to identify their fellow carnivores. The world encloses various noise/sound levels that are emitted by the agents.

## IV. RESULTS

Approximately 28 full simulations were run, video recorded, documented and analyzed. Simulation time is measured in units of epochs, which could be thought of as corresponding to 33 human minutes. Experimental simulation times range from 100 epochs (2.3 a human day) to 10,000 epochs (231 human days) as seen in Figure 9. Through various parameter tweaks and simulation changes the following results were observed.

### A. *Population Collapse*

One major evolutionary component of the agents in simulation is the co-existence of both the carnivore and herbivore populations. A delicate balance between the predators and the prey is vital in preventing either of the populations, or both, from going extinct. Population collapse was a major issue in the initial simulations because this results in the simulation ending prematurely and without sufficient time to allow higher levels of intelligence to evolve.

### B. *Open Environment*

Observations reveal that extinction of an entire species occurs particularly in the beginning of the simulations, when the agents are less evolved and naive. The weights of each agent's neural network are initially random, so the survivability of either species in the beginning is a function of chance. To prevent the common issue of complete extinction a population threshold is implemented to trigger the addition of randomly initialized agents into the environment post-inception. Its effects are seen in Figure 5. A secondary objective then becomes to reduce or eliminate the use of the threshold trigger.

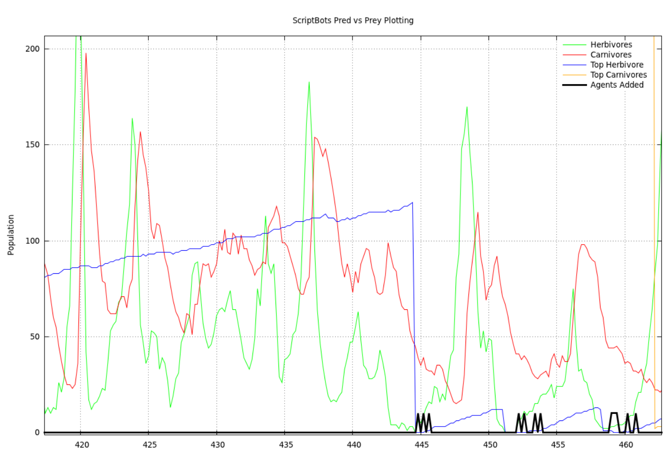


Figure 5. Population collapase of herbivore population (green line) with artificial addition of agents identified with small black lines at bottom.

### C. *Predator-Prey Dynamics*

Further investigation of the extinction issue reveals several important insights. The population dynamics generally follows the standard harmonic motion of predator-prey cycling, as shown in Figure 6. However, the standard Lotka-Volterra [8] equations (Equations 1 and 2) of population co-existence do not continuously fit the experimental data because of their assumption that genetic adaptation is sufficiently slow. In the ScriptBots simulation the probability of significant reproductive mutations is high, resulting in the occasional collapse of the harmonic cycles and disruptions in the agents' delicate balance. Lowering the probability of reproductive mutations would reduce the disruption of the predator-prey dynamics and stabilize the population. However, it would also significantly increase the time required for the agents to evolve any significant adaptations and require much larger amounts of computation.

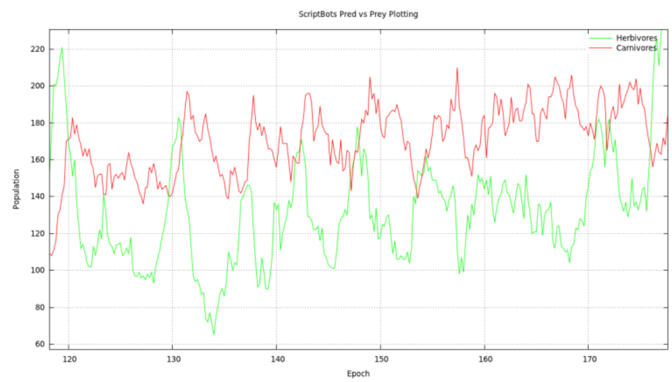


Figure 6. Fluctuations of predator (top line) and prey (bottom line) poplulations between epoch 120 and 180. Harmonic motion demonstrated.

$$\frac{dx}{dt} = x(\alpha\text{-}\beta y) \qquad (1)$$

$$\frac{dy}{dt} = -y(\gamma - \delta x) \qquad (2)$$

Equations 1 & 2. Lotka-Volterra (predator-prey) equations typically describes the interation of the population of two adversarial species.

### D. *Suboptimal is optimal*

Occasionally either the predators or the prey evolve particularly optimal adaptations in hunting or escaping one another. This species would have a large competitive advantage and quickly over take the disadvantaged species in population and fitness. The result would be the sudden decline or collapse in the other species. In particular, when the carnivores had the upper hand, they would quickly over-hunt the herbivores and cause the herbivores' population to suddenly decline or collapse. Consequently the more optimal carnivores, which rely on herbivores for sustenance, would then starve and their genetic advantage would be lost with their extinction. Thus, it is observed that in the evolution of agents for optimal co-existence, sub-optimal is actually optimal.

*E. More complex brains and stochastic behaviour*

In an evolutionary simulation where agents make decisions based on the weights of their neural networks, giving agents a higher level of intelligence is a very indirect task. Tools and abilities can be given to the agents to possibly use, but it up for them to evolve the ability and insight to use them to their advantage.

As described in section III, C. Computation on the agent's brain layout, primitive memory and planning capabilities are given to the agents to use in addition to their recurrent neural networks. While the memory is limited to only one brain tick in the past and planning limited to one brain tick in the future, observations indicate that agents behave with more complex behavior, following less predictable decision patterns in response to sensory input and demonstrating more stochastic behavior.

Another attempt at reducing the agents' simplistic input response is the addition of a random input bit into the neural network that alternated random values at random intervals. A large number of young agents at the beginning of the simulator utilize this bit but in general the more evolved and capable agents eventually lose any significant use of this bit within their brain. It seems that within the simulation the optimal evolutionary response is more deterministic in nature.

*F. Temperature Preference*

The effects of an environmental temperature gradient along the horizontal axis with agent species each having a particular temperature preference were observed. When an agent is not within their optimal temperature range they experience "discomfort" and loose a small amount of health. The intention of this feature was to encourage the segmentation of the agent population into distinct species that independently evolve their own adaptations. With a more diverse set of adaptations it is hypothesized that more interesting and intelligence behavior could emerge.

Finding the best health cost for this environmental discomfort proved problematic. When the cost for being in the incorrect temperature is set high, agents clearly separate into sub-species at different parts of the world. However, no further interaction outside their section of the environment takes place and the heavy toll on their health makes their survival difficult. A reduction to the health cost generally resulted in the agents completely ignoring this factor of the simulation and no clear division of species.

*G. Swarm Emergence*

The emergence of swarms is achieved in the simulations in a variety of ways and due to a variety of reasons.

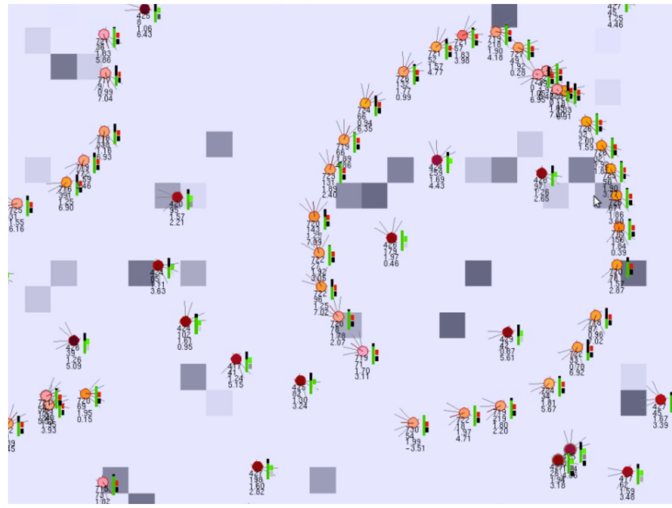

Figure 7. Example of bots forming swarm grouping in ScriptBots simulator

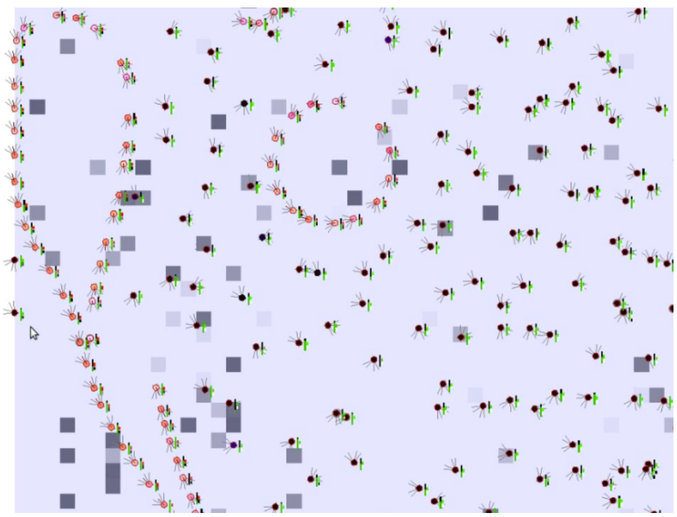

Figure 8. Example of carnivore swarm "farming" left side of screen, where all herbivores have been trapped and consumed.

Hunting packs of carnivores circling in a group frequently form because of the simulation rule that all killed prey are shared equally between all nearby predators (Figure 7). The higher the density of carnivores in a group, the higher the likely hood that a spike will correctly impale an herbivore. With this probabilistic advantage the carnivores usually evolve the ability to navigate based on their left and right eye sensors and to follow the carnivore in front of them.

Larger snake-like formations that dynamically expand and contract in size is a more advanced evolution that occasionally develops when the eye-seeing distance is increased or when a health benefit because of proximity is introduced. In the former modification it is conjectured that the agents have a longer range of vision and so are less likely to lose the agent they are following in front of them. With better vision agents are more adept at following complex snake movements that require sharper turns and occasional gaps in the structure. This

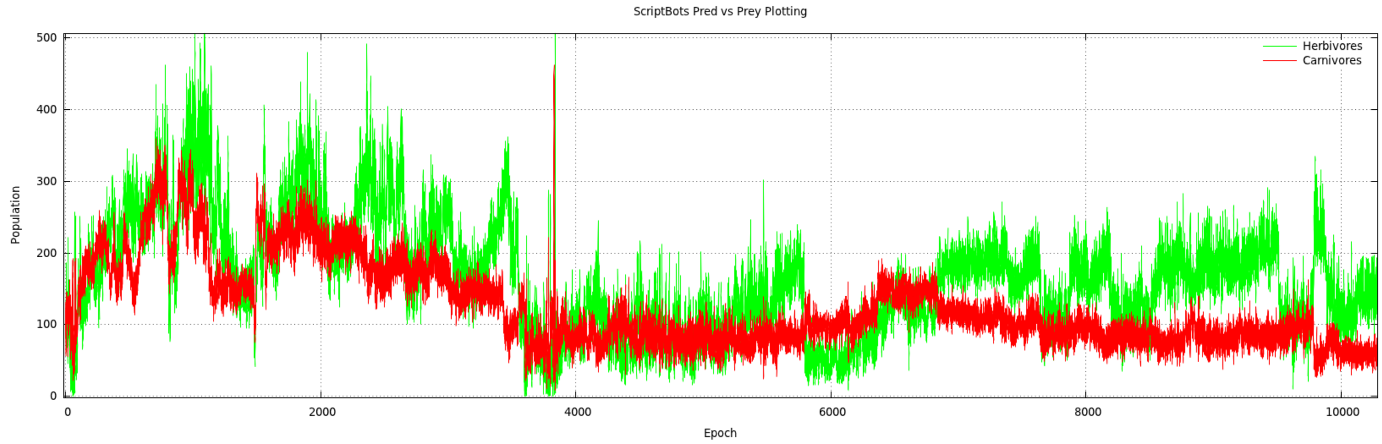


Figure 10: Final simulation of 10,000 epochs, equivalent to roughly 231 human days. Only one population collapse occurred, around epoch 3800.

behavior demonstrated surprising redundancy when two different snake chains collided, individual carnivores wanted to join or when the snake chain curled into itself. The swarm structure was able to split into sub-snakes and reconnect at other locations at will.

In the latter reason for the formation of snake-like formations, a positive health benefit is activated whenever more than one bots are within almost-touching proximity of each other. This feature can be thought of as modeling drafting (drag reduction in a liquid) or kleptothermy (huddling to share heat). While this feature of the simulation did show small amounts of influence on the agents, in general the bots did not maintain close proximity for significant amounts of time at the level of health reward offered. If the health reward was increased further the agents lost motivation to do anything more than sit stationary on top of each other for survival.

The most advanced swarm behavior observed was the emergence of herbivore "farming" (Figure 8). Long snake-chains of carnivores systematically would enclose a sub-population of herbivores and slowly shrink their swarm chain, pushing the herbivores together until they were trapped. The carnivores would then be able to effortlessly kill the herbivores with very few able to escape.

## V. FUTURE WORK

There are many areas of the simulation that remain to be investigated and potential changes that could improve the overall results of this paper's objectives.

### A. Features

The addition of greater amounts of long-term memory in the agents' brains would enable learning from past experiences and planning of future actions. Adding a small amount of noise to the outputs of the agents would add a new level of realism to the simulation and the agents' experience. Giving the agents' neural networks the ability to "learn" throughout their lifetime via weight modification would allow childhood development and learning. Addition of other motivating factors beyond avoiding death, such as pain and happiness, would give agents more survival abilities. These factors could also potentially give the agents more thoughts into their own state. This could result in a reward-mechanism by which the agents could be more motivated towards optimizing their survival.

Having obstacles and other artifacts in the simulation world would encourage more complex problem-solving behaviors. The addition of a shared resource that both carnivores and herbivores need for survival, such as water, could add another level of competition between species. The reduction of predator-prey attack impact on health would cause herbivores to lose health more gradually and give them a chance to learn how to defend themselves.

### B. Computation

An increase in the amount of computation time would significantly help the simulations. As discussed in Section V Part C the reduction of the mutation rate would better stabilize the populations and reflect more real-world evolutionary patterns. This would in turn require much higher amounts of computation and simulation time. A larger simulation world that could support more agents would results in more biological diversity. Agents with more neurons and connections in their brains would enable agents to learn more complex behaviors.

## VI. CONCLUSIONS

In this paper it has been demonstrated that teaching agents through evolutionary neural networks in a simulated environment can lead to the development of primitive intelligence and emergence of swarm behavior. These swarms have evolved the ability to perform group tasks and behaviors that would be impossible for a single agent to accomplish. Small changes in a simulated environment and its fitness function have a powerful ability to coerce the development of agents' behaviors. It has also been observed that a predator and prey population can coexist for extended periods of time but in a confined simulated environment, with a high probability of mutation, occasional population collapses seem inevitable. It has been shown that herbivores and carnivores are able to co-evolve in an arms race of offensive and defensive skills. Overall it has been demonstrated the feasibility of emergent swarm behavior and intelligence through simulated evolution.

## ACKNOWLEDGMENT

Thanks to Andrej Karpathy for the initial ideas for the evolutionary simulator and periodic guidance. Thanks to Dustin Reishus for his inspiration on the subject of swarm robotics.

## REFERENCES

[1] G Leach, John. "Survival Psychology". NYU Press 1994 and Survival Psychology.

[2] E Bengtson, S. (2002). "Origins and early evolution of predation". In Kowalewski, M., and Kelley, P.H. (PDF). The fossil record of predation. The Paleontological Society Papers 8. The Paleontological Society. pp. 289–317. Retrieved 2007-12-01.

[3] Available online: http://gbb.eldoc.ub.rug.nl/FILES/root/1982/MolCelEncrv Haastert2/1982MolCellEndocrinolvHaastert2.pdf.

[4] Andrej Karpathy, ScriptBots, Open source simluation for Evolutionary Artificial Life Simulation of Predator-Prey dynamics, Standford University, 2011.

[5] Segal, Mark, Kurt Akeley, and Chris Frasier. "The Opengl Graphics System: A Specification". Mountain View, CA: Silicon Graphics, 1992. Print.

[6] Available online: http://physicsarchives.com/index.php/courses/663.

[7] Available online: http://physicsarchives.com/index.php/courses/663.

[8] Goel, Narendra S, Samaresh C. Maitra, Elliott W. Montroll, Elliott W. Montroll, and Elliott W. Montroll. "On the Volterra and Other Nonlinear Models of Interacting Populations". New York: Academic Press, 1971