Evolving Grounded Communication in a Navigation Task using NEAT

Kristen Allen & Peter Ballen

May 11, 2012

Abstract

The independent development of grounded communication systems will be an important task for multi-agent problem-solving systems. This project attempts to reproduce results by Marocco and Nolfi (2002) on the collaborative creation of a symbolic system of communication by agents solving a navigational task. The task provides limited sensory inputs to require the robots to explore their environment and only indirectly rewards communication, which will be the robots' strongest tool in task completion. The neural network controlling each team of robots evolves to allow the strongest to mutate and reproduce. Aiming to test the strength of a more mutable network topology instead of the fixed network structure used by in the original study, we began by using the NeuroEvolution of Augmenting Topologies (NEAT). Variations of many of the parameters are discussed, including several that attempted to match our experiments to the original more closely–for different runs, we increased the population size, gave the population a longer period of time to evolve, tested an exponential activation function, and fixed the network topology to be closer to Marocco and Nolfi's version. Throughout all of these trials, results still failed to surpass baseline fitness. A later implementation focusing on a fixed network topology with elitist reproduction and no speciation resulted in similarly low-performing neural networks–with almost no variability in fitness levels, unlike our NEAT results–raising the possibility of other issues in task implementation. Ultimately, neither NEAT nor a standard genetic algorithm reproduced successful task completion or the creation of a signaling system.

1 Introduction

The problem of artificially developing communication systems in a relevant, grounded context is one which contrasts with the standard language-creation automations. Because of the nature of tasks assigned to robots, humans are unlikely to be able to invent the most useful signaling conventions–or even meanings to be signaled–for multi-agent problem-solving systems. It is thus desirable for agents to develop their own systems to categorize current situations in a way that is meaningful both to their capacities and to the task at hand, and to convey this categorical perception to each other to improve their performance while simultaneously evolving more general task proficiency.

1.1 Related Work

The problem of developing communication systems in a relevant, grounded context is one in sharp contrast to the standard language-creation automations. Because of the nature of tasks assigned to robots, humans are unlikely to be able to invent the most useful signaling conventions for multi-agent problem-solving systems. It is thus desirable for agents to develop their own systems to categorize their current situations in a way that is meaningful both to their capacities and to the task at hand, and to convey this categorical perception to each other to improve their performance while evolving more general task proficiency.

Many proposals and exploratory studies tackle the development of language by interacting robots (e.g. Steels Kaplan, 2002; Breslow, 2010; Bell, Pace Santos, 2009; Marocco Nolfi, 2002; Cangelosi Parisi, 1998). Paul Vogt notes that attempts to induce artificial language in robots may consider any of phonology, the creation of shared symbolic systems, the development of syntax, the continued evolution of word meanings, and many other issues besides (Vogt, 2006). The current study bypasses questions of sound and grammar to focus on the creation of useful single-symbol utterances. Another study demonstrates robots' capacity to form categories using language: Steels and Kaplan (2002) successfully show AIBO robots' ability to group objects in the environment into categories during a language game. Their study, however, is dependent on human feedback and is not rooted in any subsuming task. A more relevant study to the creation of useful robot language should allow the agents to develop the communication system on their own, and to ground it in their environments in order to solve a difficult task.

Marocco and Nolfi describe a task centered around robots categorizing their environments in order to perform a navigational task requiring both individual exploration and joint communication (2006, 2007). The authors aim to reward successful language use indirectly, through improved task performance. The task places four genetically identical robots into a simulated environment with two target areas; the robots aim to end up equally distributed between these areas, and are rewarded for placement on targets but penalized for clumping on one or the other. The robots have eight infrared sensors that detect objects within a very close range (less than their body length), binary target sensors, and four communication sensors that can hear up to several times their body length to the front, left, right and back. The population of individuals evolves over time using mu-lambda selection: the most successful fifth of the 100 individuals tested in each generation are replicated, with a small chance of mutation, five times each to form the new generation. Their bodies are static, as are their neural network topologies, which are fully connected between layers with three hidden nodes; an individual's genome is determined solely by its connection weights. Ultimately, the authors find that the robots can indeed cluster information about their environments into about five categories, using signals to communicate exploration, placement on a target, and even to elicit response from other robots. Since the communication sensors are much more sensitive than the infrared, robots were able to hone in on target areas by following signals from other robots that were already there. Thus, successful communication in a navigational task has been demonstrated, and was our inspiration for attempting an implementation with a complexifying network topology.

One potential criticism of Marocco and Nolfi's study is that the control–an identical system without communication sensors–was essentially operating blind, as its obstacle sensors were weak and target sensors useless until actually on the target. Thus, greater success by communicating systems reflects the utility of communication in difficult tasks, but also is somewhat expected given the myopia of the control robots. Successful use of the signal producers and receivers by the neural network, though, was still entirely the responsibility of the robots, so these signaling devices made higher fitness possible but did not guarantee it.

Our approach uses the physical grounding principle (Brooks, 1990) to motivate the creation of a basic communicative system that is a tool in a particular task, rather than being manipulated in the abstract. In some artificial intelligences, "the creator gets to tell the program the facts in some sort of representation language" (11); here, the goal is to allow the robots to develop a language that is relevant in the context of an external environment. Hence, in this experiment, like in Marocco Nolfi's paper, the experience of a robot with limited sensory capabilities is imitated as closely as possible. Sonar inputs are relatively constrained in range, but are faithful to the function and positioning of the Pioneer robot's sensors. Communication signals are sent equally in all directions (as if the robot were speaking directly upwards), and communication sensors receive information from up to one robot in each of four relative quadrants–mimicking direction-sensitive hearing. Target sensors are limited to a binary value, imitating an environment in which resources can only be sensed within a certain range. In contrast to Marocco and Nolfi's paper, though, we aimed to approach the problem with a flexible neural network, in order to enable the exploration of more divergent strategies.

In another attempt to implement communication with neural networks evolving through NEAT, Breslow (2010) created a different two-part task which required a group of robots to first collide with a red object and then, once each robot had touched it, higher fitness was granted when more robots touched a blue object. Contact with the blue puck before all robots had touched the red, though, was harshly penalized. This task is another one that would be much easier with communicating agents, but Breslow's agents also failed to develop a system of communication, reaching a maximal mid-range level of success through strong individual strategies of blue-avoidance but failing to learn to seek the blue after all robots had touched the red puck. Breslow also experimented with feed-forward networks and found that communication output was most strongly related with the perception of blue objects, which was not particularly useful to other robots and was likely simply a side effect of the hidden nodes' strength in identifying blue objects in order to aid the motor controls. Success with these feed-forward networks, however, was actually stronger than in recurrent networks and occasionally yielded successful two-part strategies. Thus, it was useful for us to examine the effect of both feed-forward and recurrent neural networks, but ultimately our robots' failure to develop useful communication was consistent with Breslow's 2010 results.

1.2 Background on NEAT

We implemented NEAT, the NeuroEvolution of Augmenting Topologies, to evolve the robots' brains (Stanley Miikkulainen, 2004). NEAT begins with a small population of neural networks with fully connected input and output layers and no hidden nodes. Each network is evaluated using a fitness function. Well-performing networks are allowing to continue evolving. Over time, networks can add new hidden nodes and new connections, allowing more complicated network topologies.

Unlike a traditional fixed network structure, NEAT does not require the researcher to determine a-priori the correct number of hidden nodes. This is convenient because choosing the wrong number of hidden nodes can be disastrous. If the researcher chooses too few nodes, the problem will be unsolvable. If the researcher chooses too many nodes, the solution will be overly complicated and it will take a long time to discover a successful solution.

The xor problem is an example of a problem that illustrates the advantages of NEAT. Given two inputs A and B, the network should output 1 if either A=1 and B=0 or A=0 and B=1. The network should output 0 in any other case. The xor problem is solvable by a network with at least one hidden node.

A naive researcher might try to solve this problem with a fixed network with zero hidden nodes. Solving the xor problem requires at least one hidden node, so this approach is doomed to failure. Another researcher might try to solve this problem with a fixed network with fifty hidden nodes. This approach would work, but training a network this large would take days. NEAT would begin with a network with no hidden nodes, then quickly converge on a solution with one or two hidden nodes. In more complicated problems, using NEAT allows a researcher to avoid trying to guess the exact optimal network structure. Since creating a communication system grounded in the environment is a complicated task, and it is not obvious how many hidden nodes the network needs–perhaps it should have one hidden node for each symbol, but we also do not know how many symbols are optimal–NEAT lets us give the network maximum freedom to evolve whatever structure is most appropriate for the task.

Speciation is another useful characteristic of NEAT. When a distinct topology is created, a new species is added to the population. This species is given a small number of generations to evolve freely before it is forced to compete with other species and possibly be eliminated. This can be a powerful tool, because when a hidden node is first added, the weights feeding into and out of that node are random, and are likely to be initially unhelpful. The network is given a few generations to adjust the weights to more appropriate values before it risks being eliminated. Different topologies, in the form of different species, compete. Bad topologies are eliminated so that more resources can be devoted to good topologies.

2 Experiment

We ran our simulation in pyrobot. Four simulated Pioneers are placed in an enclosed square surrounded by walls. Each robot is controlled by an identical neural network. Two non-overlapping targets, represented as lights, are fully enclosed in the square. The goal is for two robots to sit in one target and for two robots to sit in another target. To evaluate the robot's success rate, we use the following fitness function.

Fitness = 0.5 + 0.125 * (number of robots in a target) - 0.5 * (number of excess robots in a target)

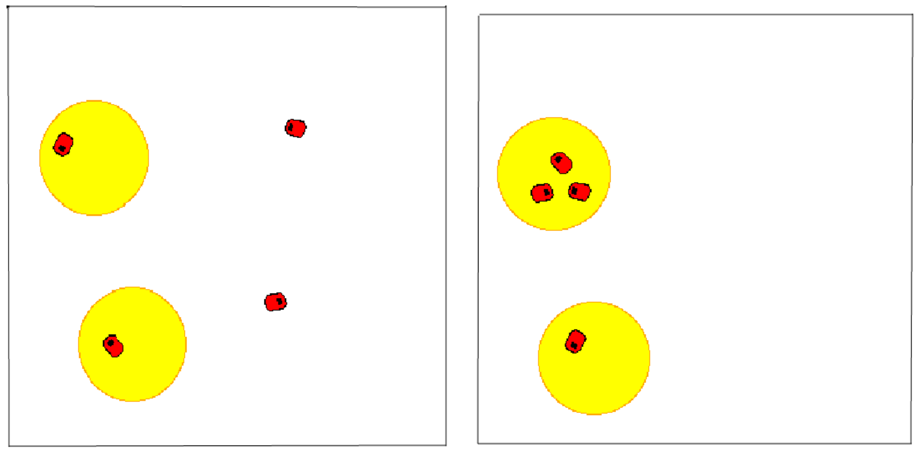


Figure 1: A snapshot of the robots in action. In the left snapshot, the current fitness is 0.75: there are two robots on a target and no excess robots on a target. In the right snapshot, the fitness is 0.5; there are four robots on targets, but a 0.5 penalty is applied because three robots are on the same target.

To gain information about the environment, each robot has eight sonar sensors, one target sensor and four communication sensors. The sonar sensors range from 0 to 1, with 0 indicated empty space and 1 indicating a wall; each sensor is the average of two consecutive Pioneer sensors. The target sensor is binary and returns 0 if the robot is not currently on a target and 1 if the robot is currently on a target. To simulate communication, the robot has a hearing radius, which is divided into four quadrants: "front", "left", "right" and "back" [1]. Each communication sensor is associated with a different quadrant. In each quadrant, the sensor uses Euclidean distance to find the closest robot in that quadrant; that robot's speech output is returned by the associated communication sensor. If there are no robots in the quadrant, the communication sensor returns 0.

[1] Quadrants are determined by the angle between the sensing robot and the sensed object. The "front" quadrant is associated with angles between 315 - 45 degrees, "left" is 45 - 135 , "back" is 135 - 225 and "right" is 225 - 315

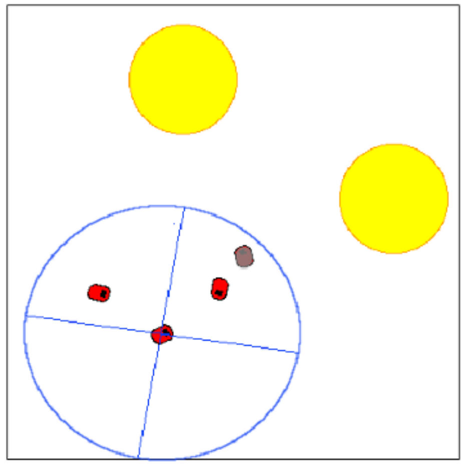


Figure 2: The robot has divided its hearing radius into four quadrants. The first communication sensor, associated with the front quadrant, returns the value being emitted by the closer non-gray robot. The second sensor returns the value emitted by the robot in that quadrant. The third and fourth sensors return 0, since there are no robots in those quadrants.

The neural network outputs three values: a left wheel motor value, a right wheel motor value, and a communication output value that can be heard by the communication sensors of other nearby robots.

To evaluate a neural network, we run three trials of 200 lifecycles each. The network's overall fitness is the average of results from the three trials. At the beginning of a trial, the robots are randomly placed in the square, where two targets are also randomly placed. At every time step, each robot calculates its 13 sensor values. Those values are given to the neural network, which calculates two motor values and a communication output. The robot will move and speak using those values. The current fitness, using the fitness function described above, it also calculated. After 200 lifecycles, the trial ends and the average of the 200 fitness values is returned. In order to speed computation, if two or more robots have stalled, the trial is terminated and the current fitness is assigned to all future lifecycles–we found that very few stalled robots ever recovered, but did not want to explicitly address stalling in our fitness function.

3 Results

The first trial we ran (the 'baseline') had a population size of 20 and ran for 50 generations. We allowed NEAT to include recurrence in the network. NEAT failed to evolve a network that could successfully solve the task. As seen in Figure 3, the average fitness at the end of the trial was 0.548, which is poor because a fitness value of 0.500 is the default value for no robots on any targets–it translates to about 9.6% fitness. The best fitness was 0.665, but this number is deceptively good. The network that achieved a fitness of 0.665 likely got a lucky initial configuration; some robots could have been randomly placed on the lights, and continued to explore in those areas. When the network was reevaluated using a random configuration, it failed to solve the task, demonstrating a lack of understanding that being on the targets granted higher fitness.

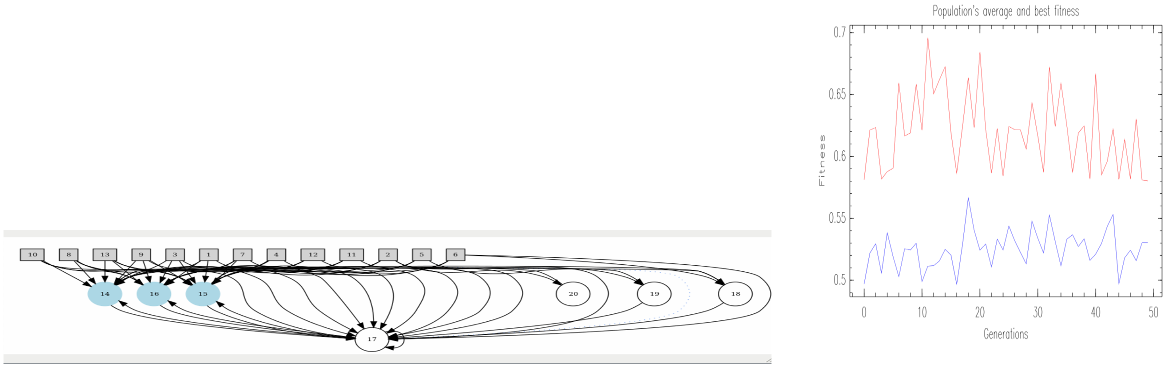


Figure 3: The neural network that evolved after 50 generations is on the left. The graph of fitness over time is on the right. The blue line represents average fitness and the red line represents best fitness.

We then tried running a trial with a population size of 100 evolving over 50 generations. A larger population offers more possibilities for mutation and crossover that could be advantageous to solving the task; candidates with good solutions have more chances of reproducing. However, even with this larger population, NEAT failed to evolve a network that could successfully solve the task. The average fitness as the end of the trial was 0.523, as seen in Figure 4. The best fitness was 0.665, but as before the network that achieved this fitness failed to reproduce the results.

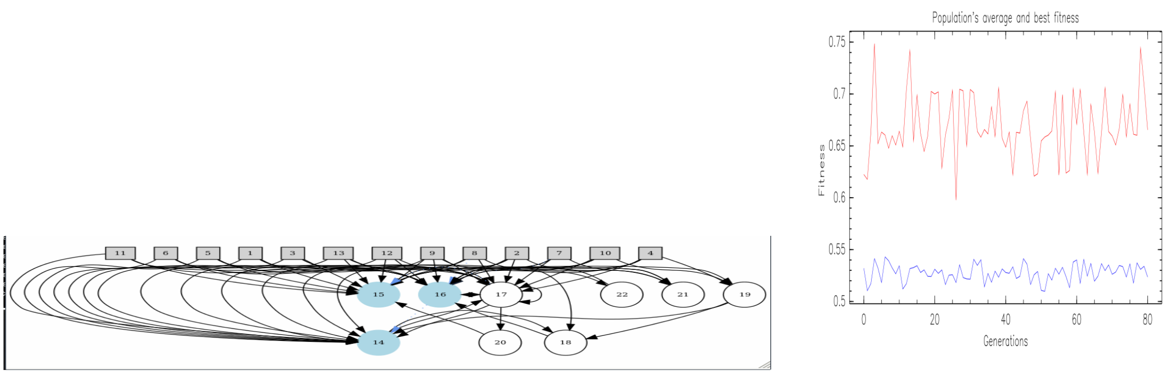
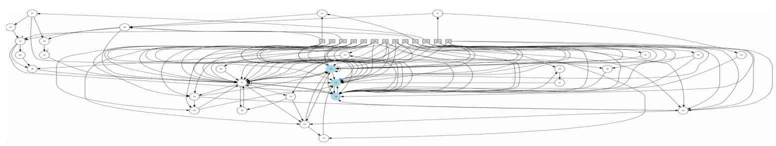


Figure 4: The neural network that evolved after 50 generations with a larger population is on the left. Notice the network is more complicated than the baseline trial. The graph of fitness over time is on the right. The blue line represents average fitness and the red line represents best fitness. Even with the larger population, there was no significant improvement in performance

We then ran a population of size 20 for 320 generations. Evolution over more generations allows NEAT to evolve more complicated topologies that might be better equipped at solving the task. Indeed, NEAT evolved a very complicated topology; see Figure 5. It's worth noting, though, that Marocco and Nolfi's robots managed to solve the task using 3 hidden nodes, implying that there are many extraneous hidden nodes in this network. However, even with six times as many generations as before, NEAT still could not solve the task.



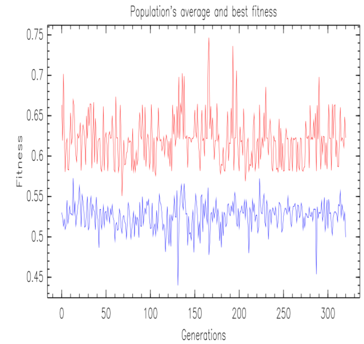


Figure 5: The network that evolved after 320 generations. The network is incredibly complicated, but there is no improvement in results

The next trial we ran used a semi-fixed topology. The network started with three hidden nodes and was not permitted to add additional hidden nodes; it was only allowed to alter the connections between nodes. By setting the number of hidden nodes constant, we hoped NEAT would be more successful at solving the task. However, this did not significantly change performance, in fact decreasing the average fitness–perhaps because they made it more difficult for robots to even develop a tendency to stay on the targets once there.

We then tried changing activation functions. For the previous trials, we used the tanh activation function. We experimented with an exponential function instead, but this did not improve performance either.

The final approach we tried was to abandon NEAT and using a fixed recurrent network with three hidden nodes that fed into each other. This network also failed to complete any part of the task; we did not have time to fully test the resulting networks, but they seem to fail even to adequately explore the environment and stumble across the targets.

Summary of Results	Average fitness	Best fitness
Baseline	0.54754	0.66479
Population of 100	0.52356	0.66542
320 generations	0.49966	0.62271
Fixed hidden nodes	0.50803	0.58333
Exponential activation function	0.50676	0.70625
Fixed network	0.50000	0.50000

Table 1: Summary of results

4 Conclusion

In conclusion, we could not use NEAT to solve the communication task. There are a few reasons why this may be the case. One explanation is that there are hidden bugs in our implementation or places where we unknowingly diverged from Marocco and Nolfi's implementation. Seemingly unimportant changes in our implementations could have impacted our results. For example, we used pyrobot as our simulator, whereas Marocco and Nolfi used an unspecified implementation (probably not pyrobot), which could have impacted our results because the robots have different defaults for motion and sensor values. Similarly, they use a host of complicated activation functions, whereas we simplified to the basic tanh and exp functions. Those authors also used a slightly smaller sonar range than we did. While we do not believe errors in our implementation of the task with NEAT networks to be the cause of the poor results, it is one possible explanation.

Another explanation may be that we didn't run the trials long enough. Marocco and Nolfi ran 2000 generations, while we ran at most 320, and more frequently 50. This criticism is unpersuasive for two reasons. First, Marocco and Nolfi managed to achieve meaningful improvement within the first 50 generations, which is why we were comfortable running most tests for only 50 generations. Even if we couldn't perfectly replicate their results, we should have seen some improvement over that time. The second reason this argument is unpersuasive is the performance of the 320-generation trial: as seen in Figure 3, NEAT clearly had enough time to evolve a complicated network structure with many hidden nodes; it just failed to create properly-weighted connections between the nodes. Thus, the lack of initial rapid improvement is more likely to be the cause of task failure.

A more persuasive argument is that NEAT is fundamentally unfit for solving this task. NEAT creates incredibly malleable network structures; nodes and connections are constantly being added, removed, and adjusted. This constantly changing behavior is not well-suited for solving a task such as exploring and navigation. The task isn't complexifying over time, so an ideal neural network would pick a (small) number of hidden nodes and focus on perfecting those weights. Instead, NEAT wastes time and population space adding new nodes and connections, rather than spending all of its resources trying to optimize its existing network to the task at hand. The failure of our fixed-topology network is discouraging because it seemed to be the best chance to replicate Marocco and Nolfi's results; the results on that trial were so close to do-nothing fitness levels that future work should begin by re-examining that code and the performance of the resulting networks.

Another potential strength of the original paper's methods is the use of mu-lambda selection instead of speciation, as is used in NEAT; elitist selection of reproducers provides less diversity in neural networks, but a greater chance that those who adopt even small positive changes in strategy will be able to pass these on to the next generation. NEAT's protection of novelty through speciation detracts from an emphasis on task fitness, which may be disastrous in work on a task whose first steps–exploration and staying on targets–are relatively simple. Ultimately, the agents' failure to complete even the initial phases of the task mean that signaling systems could not be explored; individual fitness and the implied ability to usefully categorize environmental inputs are prerequisites to communicating meaningful information to fellow robots.

5 Bibliography

Bell, A., Pace, A. & Santos, R. (2009). *Evolutions of Communication.* Swarthmore College CS Department. Retrieved March 23, 2012, from `http://web.cs.swarthmore.edu/~meeden/cs81/s09/finals/AlexAndrewRaul.pdf`

Billard, A. & Dautenhahn, K. (1998). *Grounding communication in autonomous robots: An experimental study.* Robotics and Autonomous systems 24 (71-79).

Breslow, A. (2010). *An Application of NEAT: Recurrent Neural Networks and a Communication Task.* Swarthmore College CS Department. Retrieved March 23, 2012, from `http://web.cs.swarthmore.edu/~meeden/cs81/s10/finals/Alex.pdf`

Brooks, R. A. (1990). *Elephants don't play chess.* Robotics and Autonomous Systems 6 (3-15).

Cangelosi, A. & Parisi, D. (1998). *The Emergence of a Language in an Evolving Population of Neural Networks.* Connection Science, 10:2 (83-97).

Marocco, D. & Nolfi, S. (2007). *Emergence of Communication in Embodied Agents Evolved for the Ability to Solve a Collective Navigation Problem.* Connection Science, 19:1 (53-74).

Marocco, D. & Nolfi, S. (2006). *Origins of Communication in Evolving Robots. From Animals to Animats.* In Nolfi, S. et al. (Eds.) 4095 (789-803).

Steels, L. & Kaplan, F. (2002). *AIBO's first words: The social learning of language and meaning.* Evolution of Communication, 4:1 (3-32).

Vogt, P. (2006). *Language evolution and robotics: Issues on symbol grounding and language acquisition.* In A. Loula, R. Gudwin & J. Queiroz (Eds.) Artificial Cognition Systems (176-209). Hershey, PA: Idea Group.