Title

Evolution of a Simple Compositional Language Using Animat-Based Modeling and Simulation

Author
Moritani, Masaki

Publication Date
2013

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Evolution of a Simple Compositional Language Using Animat-Based Modeling and Simulation

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

Masaki Moritani

2013

Abstract of the Thesis

Evolution of a Simple Compositional Language Using Animat-Based Modeling and Simulation

by

Masaki Moritani

Master of Science in Computer Science

University of California, Los Angeles, 2013

Professor Michael G. Dyer, Chair

Human communication has features, such as syntax, unseen in any other form of animal communication. How did we come to use such a sophisticated form of communication? This paper addresses the issue of the origin of compositionality in languages using animat-based modeling. In the simulation, 100 software agents controlled by neural networks are given the ability to communicate with each other using sequences of signals. These agents—called animats—play communication games with each other, and if they are among the most effective communicators, they mate and produce offspring. Through evolution and learning, animats developed languages that exhibit some rudimentary compositionality in simple environment settings, but under more sophisticated environments, they struggled to establish effective protocols of communication. Results suggested that simple recurrent neural networks allowed animats to develop languages with rudimentary compositionality, and that imitation learning helped animats come to consensus on usage of signals.

The thesis of Masaki Moritani is approved.

Eleazar Eskin

Adnan Darwiche

Michael G. Dyer, Committee Chair

University of California, Los Angeles

2013

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Intelligence is often attributed to language and communication skill. We humans use language with complexity that is unparalleled by that of any other animal on the planet. How did such complexity come about? What is the origin of language, and how did it evolve to forms we see today? My project examines this problem using simulations of communicating software agents with a focus on compositionality.

One distinguishing feature of human language complexity is compositionality. When we speak, we sequence phonemes into words, and words into sentences. Each human language has a limited set of phonemes, and yet we can make an unlimited number of words and sentences to express ideas. This ability to combine parts in a sequence to communicate a numerous, if not infinite, number of ideas is the main topic of this project.

1.1 Goals

In the experiment, I set up a software environment in which individual agents controlled by neural networks—called animats—can communicate and interact with each other. These animats converse with each other using signals and trade resources, and reproduce to evolve over generations to better adapt to the environment. The primary goal of this project is to develop such a simulation system so that animats can evolve to handle messages composed of multiple signals in a sequence to express more ideas than the size of their repertoire of signals.

The significance of this experiment lies in that signals are emitted in a sequence to specify a combination of things. The scope of the experiment does not include the development of complex humanlike syntax rules, such as subject-predicate relationships. However, the physiological and intellectual capacity to produce and interpret sequences of signals is an important step towards having syntax rules. What do effective multi-part sequence messages require of control structures? What prevents agents from developing such a language? How do various factors in the environment alter the evolutionary trajectory?

With these questions in mind, I have designed communication games that the animats can take part in to develop language. Using simple recurrent neural networks, genetic algorithms (evolution), imitation learning (using back propagation), and seeding, I compare the various effects of each on the evolutionary trajectory of the animats and of the signal language to look for some basic requirements of developing syntax.

CHAPTER 2

Examples of Past Works

There has been much work starting from the 1990s that looked at the problem of the origin of language from a computational point of view, many of which use simulations with agents controlled by neural networks. In this section, I will list some of these works and comment on how they may relate to my project.

Steels (2005) [Ste05] outlines concepts faced by computational approaches to the origin of language problem. Steels first describes the three aspects of language: biological capacity, language inventory, and communal language. Biological capacity means that the organism has the physical ability to produce language (e.g. vocal cord, brain). Language inventory means that there is a repertoire of symbols in the language. Communal language means that the language (semantics, syntax, etc.) is shared and agreed upon among the population.

Steels also describes seven stages of language complexity.

- The first stage involves having names for individual objects.
- The second stage involves having a name or a word for categories of objects.
- At stage 3, language allows the use of multiple words (compositionality) to refer to multiple categories.
- At stage 4, members of the population are able to produce and interpret syntactic patterns ad hoc. This means that there is no set grammar in the language, but the agents are able to make use of syntactic patterns.

- At stage 5, language adopts systematic grammar; speakers of the language are aware of the syntax present in the language.
- Steels defines stage 6 to be a point where predicates can modify other predicates as well.
- Finally, at stage 7—the natural language level—the language becomes capable of describing itself.

In terms of work by Steels, my project aims to lay the groundwork for reaching stage 4 primarily by evolution. I cannot strictly say that the goal is stage 4 itself, however; in my model, categorization is hard-coded. The decision-making logic, not the neural network, differentiates one category from another during the interpretation phase.

Arita and Taylor (1996) [AT96] designed a simple model for investigating the evolution of communication. In their simulation model, animats take part in one-to-one conversations with neighbors, discussing a common topic: a nearby object that they both can sense. During the conversation, they both sense a set of attributes describing that object, and they produce a word (a signal) for it using their 3-layered feedforward neural network. They are rewarded whenever their words match. The simulation succeeds in demonstrating how agents agree on language. At the same time, the simulation showed an interesting phenomenon where language consensus does not converge. Instead, language consensus seems to changes over time. For example, all members of the population may agree to use word A for some object X at one point in time, but there will be an individual that starts using word B for the same object, and eventually, the whole population may end up adopting the usage of word B for object X. Even though Arita and Taylor's model was simple, it succeeded in demonstrating communicational phenomena that we see in the macro.

There have been works that specifically look at emergence of simple syntax—

work by Cangelosi from 2001 [Can01] is one among such. He discusses two simulations. The first model studies the emergence of signal communication. His model gives animats the task of identifying encountered mushrooms as either edible or poisonous and taking actions accordingly. In order to do this, animats must communicate with each other whether encountered mushrooms are edible or not. Results were successful, as animats were able to evolve a simple and efficient language for communicating the quality of food. However, the language is only a signal-object association system.

Cangelosi also conducts simulations dealing with the development of syntax including a subject-predicate composition. In this model, there are edible mushrooms and poisonous mushrooms like the first simulation, but each of these two categories have 3 subcategories. Upon approaching a mushroom, the animat must first decide if it is edible or not, and if it is edible, it must identify which of the three categories of edible mushrooms it is. Fitness is given for getting the identification correct. A verb-noun language emerged in several simulation runs; animats would have a rule of specifying both eat-or-don't-eat and which of the six mushrooms it is. His language differs from the one my model aims to obtain in that these multi-part signals are sent simultaneously, not in a sequence.

Kenny Smith's work from 2002 [Smi02] focuses on cultural transmission of language using a simulation more dependent on learning than evolution. Smith used agents with two-layered bidirectional neural networks that have a learning period when they are born. During this learning period, an agent can observe the population's signals and their corresponding meanings in order to learn the language. In this case, learning means updating weights on the neural network. Weight update rules are chosen from 81 that he devised; Smith studied their effects on cultural transmission of language. By comparing the weight-update rules, he concluded that there exists a hierarchy of learning rules, and that optimal (or near-optimal) communication is affected by learning biases. The interesting feature of this study is that it puts more emphasis on learning than evolution or innate features.

CHAPTER 3

Design Details

3.1 Assumptions About Animats and the Environment

This project investigates the evolution of language. To do this, many processes in the world are either simplified or abstracted so that the simulation can focus on communication. This section explains the basic assumptions about the simulated world.

First of all, let us define animats to be software agents living in the simulated environment. Animats are controlled by neural networks. In the experiment, each animat possesses two neural networks and a genome for encoding the neural networks. Each of these networks is a simple recurrent neural network. Later sections describe in more detail the networks and how the animat genome works. Animats have the ability to produce resources, but they cannot consume the resources they've produced. Instead, they must trade and consume resources produced by other individuals.

The animats' interface to the outside world consists of a signaling mechanism, sensors for signals, and the ability to offer a resource. This offering action is used in the communication games, in which animats will trade resources, and gain fitness whenever a correct offer has been made. The animats are not divided into male or female; every other animat is a potential mating partner, but they will not mate unless they can demonstrate effective communication.

Animats live in a world in a population that communicates with each other using signals. The concept of physical location or direction is abstracted away; there is no concept of which animat is where. Also, the concept of time is simplified into time steps, stages and generations or rounds. Each round is one generation of the animats. Each generation goes through stages of birth, learning from parent (if applicable), resource obtaining (by playing communication games), and reproduction. When a conversation occurs, it is a one-to-one conversation. The animats send each other a sequence of signals in a fixed number of time steps and trade resources accordingly. The rules of the conversation game are discussed in the next section.

3.2 Game Rules

Each animat has a vocabulary of V signals, which will be used to describe items. There are N categories of items, and each category consists of M subcategories; therefore, there are $N \times M$ different items in the world. Categories can mean something like "meat" and "vegetable." Subcategories can mean something like "beef," "chicken" and "pork" in the case that their category was "meat."

V, N and M must be set correctly in order to design a world in which animats must develop multi-part message communication in order to succeed. In the simulations, I made it possible for the animat vocabulary to cover all items, so that one signal could correspond to one item. However, at the same time, I made it their task to communicate a combination of items to each other. If the vocabulary is not big enough to cover all possible combinations of items, animats must develop another way to express all the combinations. Therefore, the size of animat vocabulary is determined by $N \times M \leq V < M^N$. I simply used $V = N \times M$. The simulations use $N = 2, 3, 4$, and $M = 3$. For terminology, call the category-n subcategory-m item "c<n>s<m>" so that, for example, category-1 subcategory-2 item would be called item "c1s2."

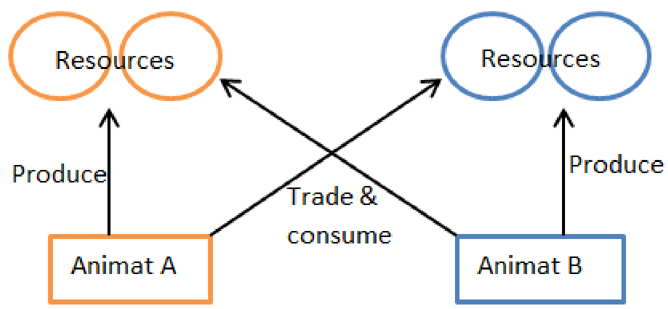

Figure 3.1: Relationship between two animats in a communication game.

We want a situation that is complex enough so that it encourages animats to develop a more sophisticated communication system. However, it is unlikely in Nature that such a complicated task arises out of nowhere. It assumes that the population experienced simpler tasks before they became complex. This is why the animats will play simpler "communication games" before they become more complex. There are two types of games played: a simpler "naming game" played by the first generations, and a more complex "combination game" played by the latter generations.

Each animat has the ability to produce resources, but they cannot eat these self-produced resources themselves. However, they are able to eat resources that are produced by other individuals. Thus, these animats must trade for each other's resources.

The first game—played during the first 400 rounds—requires animats to name one resource of request, and also correctly respond to their partner's request for one resource. Our population of simple animats is equipped with the ability to trade resources (items). These animats, upon finding another individual, strike up several trade deals. Each trade will be one resource for one resource. Animats are given N time steps to communicate with the trading partner and to decide what to offer to the partner. Animats send 1 signal per time step. If there are X number of signals, then the signals are $\{0, 1, 2, ...X - 1\}$. Each animat feeds the request to its production neural network for N time steps to produce a sequence of signals, or, a message, of length N. Then, each message will look something like "0000" ("0" signal repeated 4 times) or "1313" ("1" signal followed by "3," then repeat), if the message length was 4 signals long. These messages are in turn interpreted by the partner animat using its interpretation simple recurrent neural network. If the animat can offer the correct resource to the other animat, then it can receive what the other animat offers and eat it. Set the fitness points gained by offering the correct resource to be 50 points. If the animat offers the wrong kind of resource but gets the category correct (e.g. an animat is requested chicken but offers beef), the animat gets a portion of the resource. Set the fitness points gained by this to be 25 points. This is repeated 50 times (50 trade deals) for each pair. Even if an animat does not get the requested resource, the received resource still has nutritional value to the animat that receives it; thus, it is more important to offer the correct item than to receive the requested item. The goal of animats for this naming game is to learn to name all types of resources; by the design of the simulation, this most likely means that animats will associate one type of resource with one signal from their vocabulary.

The second game, played in the latter 800 rounds, is similar to the first game, except that the animats must now request and offer N items of different categories for each trade. While animats in the first game can only offer one item per trade, animats in this game can offer up to N items.

3.3 Simulation Dynamics Design

The simulation starts in the first round with all animats having random genotype and phenotype, unless seeding is in effect for the simulation run. If seeding is in effect, 20 out of 100 animats will begin with a preset genotype in the first round. Each round consists of 3 stages: birth, communication games, and selection/reproduction, plus an imitation learning stage after birth if enabled. Animats engage in communication games to gain fitness and to find best partners. The best pairs in each generation will mate to produce the next generation. Read the section under "Implementation Details" for more details about learning, selection and reproduction.

Data must be gathered to visualize the language the animats are using and the evolutionary trajectory of animats themselves. For this purpose, I have recorded:

- Fitness of each animat for each round. This measures how well-adapted the population as a whole is to the environment, since each animat must communicate with every other animat in the population.
- Combined fitness of each animat pair for each round, measured per pair. This measures the effectiveness of communication between animats; I will refer to these values as communication effectiveness from this point on.
- Intended meanings, the corresponding signal messages, and how they were interpreted. This identifies patterns in signal production and interpretation.

CHAPTER 4

Hypotheses

The main hypothesis of this experiment is that animats that are aware of their own signals emitted previously are able to evolve to produce and interpret messages that take advantage of sequencing. Given an environment in which the vocabulary of signals does not cover all ideas to be expressed, the animats should develop a method to overcome this limitation using composition. However, the structure of the neural network may relate to the complexity of language that the animats become capable of producing. A simple recurrent neural network with one context layer will only remember a state one time step previous to the current time; therefore, each signal is only dependent on current input and one previous state. This may hinder animats from finding effective communication methods for cases when the number of categories increases.

CHAPTER 5

Implementation Details

5.1 Neural Networks

The neural networks that control the animats are simple recurrent neural networks. Networks are composed of 3 main layers and 1 context layer. The input layer neurons directly take input from sensing the environment or internal states. Input is fed into the hidden layer neurons. Each input going into the next layer neuron is multiplied by weights assigned to each connection. Values coming out of the hidden layer neurons are then fed into context neurons and output neurons. Context neurons store the values of hidden layer neurons for one time step and feed the value back to the corresponding hidden layer neuron in the next time step. The output layer neurons take values from the hidden layer, multiplied by weights, and output values that decide what action the animat will take in that time step.

All neurons output a floating point value between 0 and 1 using a sigmoid activation function, except for context neurons. Context neurons simply store the previous value of the corresponding hidden layer neuron and directly output it back into the hidden layer. The output O of each neuron that uses a sigmoid activation function is calculated as follows: $O = \frac{1}{1+e^{-x}}$, for $x = \sum_{i=1}^{n} a_i w_i$, where a is an input value, n is the number of inputs to the neuron, and w is its corresponding weight. x, the weighted sum of all inputs coming into the neuron, is called the activation of the neuron.

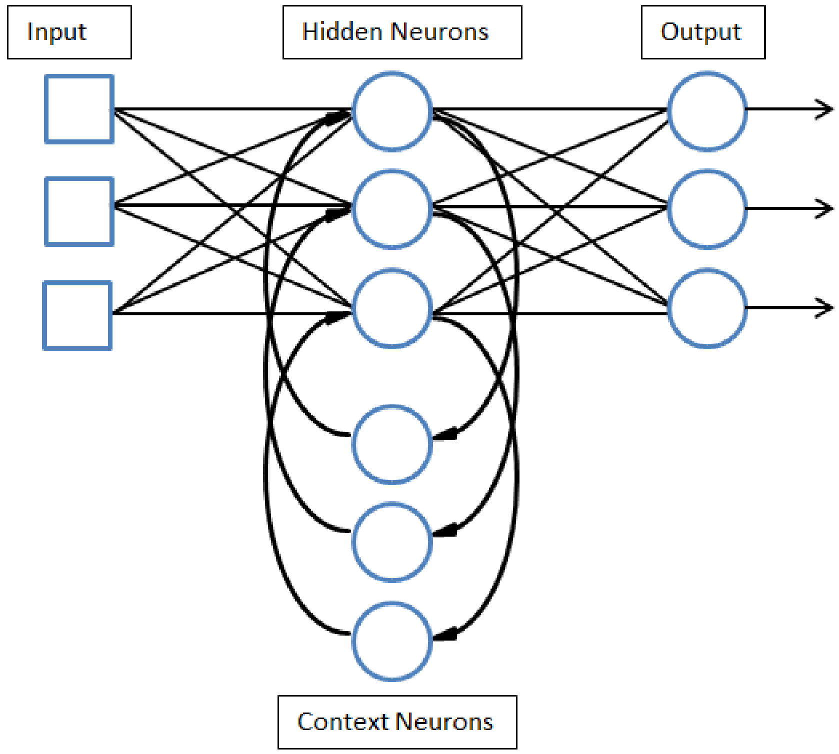

Figure 5.1: A simple recurrent neural network with 3 inputs, 3 hidden layer neurons, 3 context layer neurons, and 3 output layer neurons. Lines and arrows show connections between neurons.

Each animat is equipped with two such neural networks. The production neural network takes the item (or items) of request as input and outputs one communicational signal addressed to its partner. Which signal to send is decided by a winner-takes-all policy: the signal corresponding to the output neuron with the highest activation is chosen. The number of hidden neurons is the same as the number of output neurons, which is equal to the number of signals. The interpretation neural network takes input of which signal has been received and outputs a corresponding action (add an item or change an item in the list of items to offer). This decision is also made by a winner-takes-all policy.

5.2 Genetic Algorithm

Each animat is equipped with its genome binary string that encodes for weights on connections in its neural networks. Each encoding for one weight consists of 7 bits: 1 sign bit and 6 mantissa bits. Of the mantissa bits, 2 bits are to the left of the floating point, and 4 are to the right. The weight encoding thus has a granularity of 2^{-4}. In figure 5.2, "0011010" encodes for a weight of 0.8125 because the bits worth 2^{-1}, 2^{-2} and 2^{-4} (third, fourth and sixth bits respectively) are 1, and $2^{-1}+2^{-2}+2^{-4} = 0.8125$. Similarly, in figure 5.3, "1001101" encodes for -2.375 because bits worth 2^{1}, 2^{-2} and 2^{-3} (first, fourth and fifth bits respectively) are 1, and the sign bit (last bit) is 1, so therefore $-1(2^1 + 2^{-2} + 2^{-3}) = -2.375$.

How animats are selected to reproduce has effects on evolutionary trajectory. The genetic algorithm depends on (1) how animats are selected for reproduction, (2) how they are paired, and (3) how much mutation occurs during crossover. The selectional pressure applied to the population is communication effectiveness: animats are evaluated by pairs on how effective the communication has been. The top performing pairs each produce 4 offspring animats. Since there are 100 animats in each round, 25 pairs of animats are selected for reproduction. Having 4 children per pair makes it easier for the population to agree on a common language quicker. The mutation rate is 1.5 % ; this is helpful in producing new genes that may give the animat an advantage over others and ultimately benefit the whole population. Also, inheritance is Mendelian and not Lamarckian; learned attributes are not passed on through genetics.

5.3 Imitation Learning

Imitation learning occurs by using a parent's outputs as correct examples to perform backpropagation on the child. More specifically, the type of backpropaga-

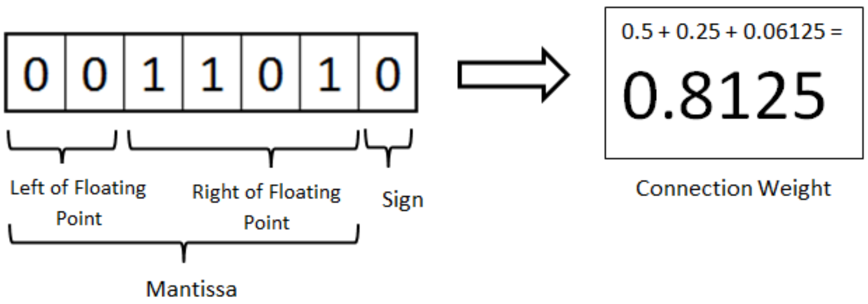

Figure 5.2: The process of translating the gene binary string into a connection weight.

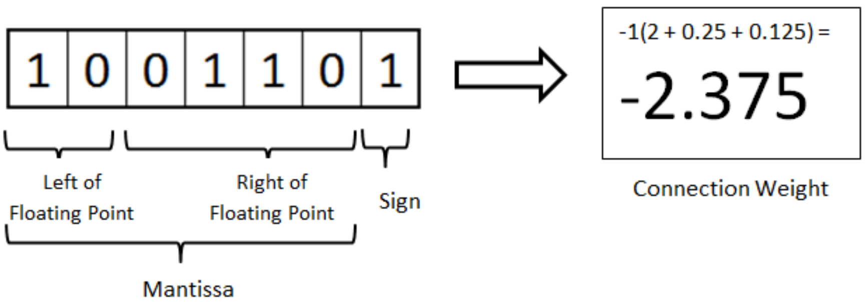

Figure 5.3: Another example of translating the gene binary string into a connection weight.

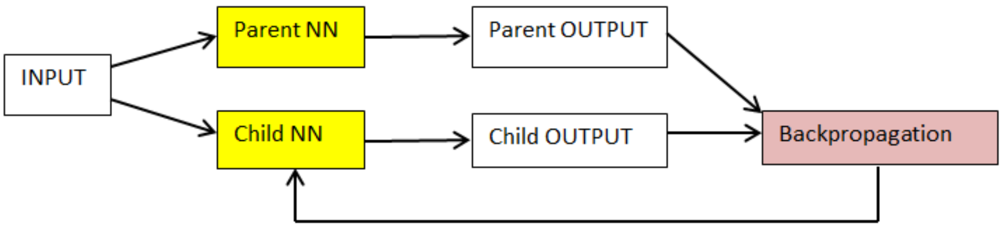

Figure 5.4: Mechanism of learning by backpropagation. Children learn by comparing their outputs with its teacher's (parent's) output and trying to correct the neural network that produced them.

tion algorithm used is called backpropagation through time [Wer90]; it is a form of backpropagation that can be applied to recurrent networks. To do imitation learning on the production neural network, a child and its parent are given the same input, and the child's output signal is compared to the parent's output signal. The discrepancies are then propagated backwards to modify weights on neural network connections. To conduct imitation learning on the interpretation neural network, the parent makes a request using its production neural network and the child responds using its interpretation neural network, and the child's output is compared to what request the parent had meant to make. When this learning process occurs, the parent and child animats go through 200 example inputs. To account for the possibility that the child perceives the parent output wrongly, the observed parent output will be wrong 10 % of the time.

5.4 Seeding

One way that the neural networks are connected is that each item corresponds to a signal (call this state "mapped"). Animats that are mapped associate each item with a specific signal at birth; in other words, if two seeded animats communicate with each other, they both are pre-equipped with the same item-signal associations, so they will get a perfect score in naming games. The question, however, is whether this neural network can adapt to the combination games to produce and interpret more complex messages that require more than perfect item-signal associations. These seeding experiments assume that there are animats with optimal ability in the naming games; for this reason, they will skip the naming game rounds and go straight into combination games.

The way “mapped” seeded animats work is the following. In the production neural network, the number of neurons each layer has is the same: there are as many input neurons as there are hidden neurons or output neurons. By mapping 1-to-1-to-1 each neuron in the input, hidden and output layers and giving a weight of 1 to each of these connections and a weight of 0 to any other connection, each input neuron will only excite its corresponding hidden and output layer neurons. Similarly in the interpretation neural network, since all layers have the same size, each neuron in each of the input, hidden and output neurons can be associated using a weight of 1 in their connections, and making all other connections to be 0. This way, these seeded animats have a built-in association between items and signals in both production and interpretation. When this is the case, all of these seeded animats produce the same signal in response to the same item, and they also interpret the same signal as the same item. As a result, when the task only involves communicating one item (i.e. naming games), the seeded animats can perfectly communicate with each other, understanding each other’s intentions. For example, if seeding makes a “mapped” animat produce signal X for a category-A subcategory-B item, then all of the other seeded animats will also do the same, and they will interpret signal X as category-A subcategory-B item. Thus, these seeded animats will be able to obtain perfect scores in the naming games.

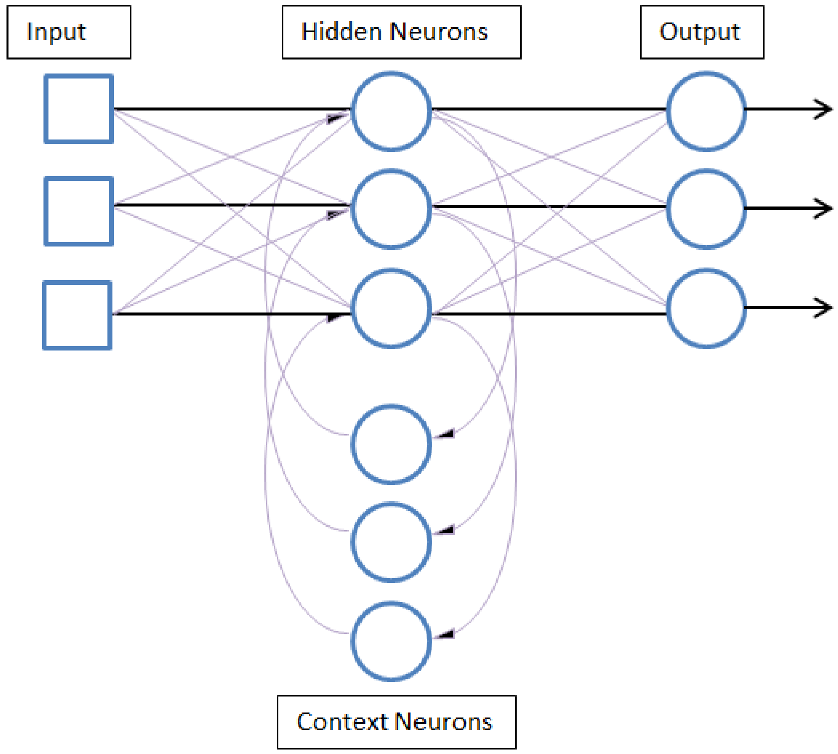

Figure 5.5: A "mapped" simple recurrent neural network of a seeded animat. All black and thick connections in the figure have a weight of 1. Any other connections—drawn with thinner purple lines—have a weight of 0.

CHAPTER 6

Results

6.1 World Complexity and Language

I ran simulations with number of categories $N = 2, 3, 4$, and number of subcategories $M = 3$. By altering the number of categories and subcategories of items, I can increase or decrease the complexity of the world and therefore control the level of language complexity needed to optimally fulfill the task. Increasing the number of subcategories will increase the number of items and vocabulary only; increasing the number of categories will increase not only the number of items and vocabulary but also the message length.

Figure 6.1 shows the communication effectiveness—measured in fitness points per pair—for the $(N, M) = (2, 3)$ case. The first 400 rounds are naming game rounds, and the following 800 rounds are combination game rounds. The maximum possible fitness points for each pair in the naming game rounds are 5000. This is because each pair plays 50 trade deals, and each trade gives each animat a maximum of 50 points, and the total fitness gained by each animat of the pair during the 50 trades is combined, so (50 points $\times$ 50 trades) $\times$ 2 animats = 5000 points. Similarly, the maximum possible communication effectiveness points in the combination game rounds are 10000. This is twice as much as the maximum obtainable communication effectiveness for naming games because each trade can reward each animat a maximum of 100 points (50 points per category, and there are 2 categories). In the graph, the red line shows the combined fitness of the top

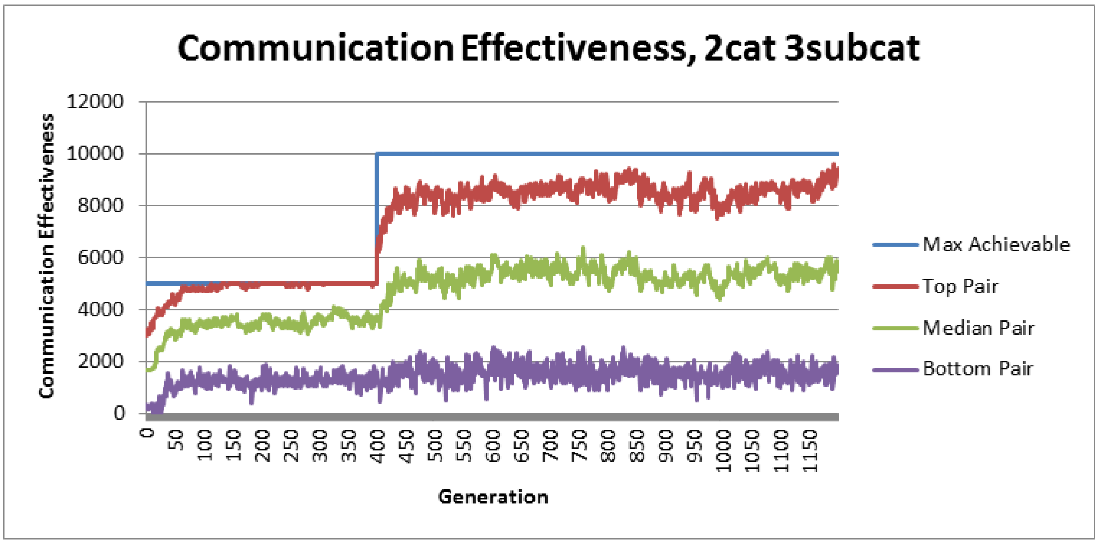

Figure 6.1: Communication effectiveness of animats when there are 2 categories and 3 subcategories of items, with random initial generation. The first 400 generations play the naming game. The latter 800 generations play the combination games.

performing pair. The green line shows the performance of the median pair, whose performance was at the median when compared with all other pairs. The purple line shows the performance of the worst performing pair.

Animats are seen to quickly make associations between items and signals. During the naming games, the top performing pairs, after about 200 generations, reached perfect communication effectiveness value of 5000. The speed at which animats evolve to associate these signals should not be surprising, as their task is only to associate 6 items with 6 signals. Even during the combination games rounds, the animats continue to do relatively well in the communication games, as the top performing pairs score communication effectiveness values of 8000-9000 when the maximum achievable value is 10000. Yet, they were not able to perfectly communicate every unique expressible idea.

Table 6.1 shows the language used by the top pair at the end of the 400

Intention	Message	Interpreted as
(0, -)	44	(0, -)
(1, -)	22	(1, -)
(2, -)	55	(2, -)
(-, 0)	11	(-, 0)
(-, 1)	33	(-, 1)
(-, 2)	13	(-, 2)

Table 6.1: Messages sent to each other by the top performing pair after 400 naming game rounds. Communication was perfect; the pair received the maximum amount of fitness points achievable.

naming game rounds. The intention shows the animats' input to the production neural network. In this example, there are 6 kinds of intentions to express; 6 messages are produced by using the production neural network. The hyphen in the table means null; for example, the meaning $(0, -)$ means "item of category 0 subcategory 0 (for brevity, call the item 'c0s0') and nothing from category 1." $(-, 2)$ would mean "nothing from category 0 and item c1s2 from category 1." The 6 signals are denoted by $\{0, 1, 2, 3, 4, 5\}$. Similarly, since there are 2 categories and 3 subcategories, the categories are denoted by $\{0, 1\}$ and the subcategories $\{0, 1, 2\}$. According to the table, if an animat wants to request item c0s1 and nothing from category 1, the entry with intention $(1, -)$ shows that it would produce a message "22." This is done by feeding $(1, -)$ into the production simple recurrent neural network for two time steps; in this particular case, the network outputted the signal 2 twice, thus producing the message "22." This message is then interpreted by the partner animat using its interpretation neural network. Interpretation works in a similar fashion. Since the message length for "22" is two, interpretation takes two steps. The animat feeds signal 2 followed by another signal 2 to the interpretation neural network. At each time step, the animat picks an item to offer. During the naming game rounds, the item the animat actually offers is the item chosen in the last time step. During the combination game rounds, the set of items the animat offers is the combined set of items the animat

Intension	Message	Interpreted as
(0, 0)	44	(0, 0)
(0, 1)	45	(0, 1)
(0, 2)	43	(0, 2)
(1, 0)	22	(1, 0)
(1, 1)	25	(1, 1)
(1, 2)	23	(1, 2)
(2, 0)	55	(2, 1)
(2, 1)	55	(2, 1)
(2, 2)	53	(2, 2)

Table 6.2: Messages sent to each other by the top performing pair after 400 naming game rounds and 800 combination game rounds. There are 2 categories and 3 subcategories of items. Communication effectiveness, calculated by the pair's combined fitness points gained during their game, was 9450.

chose in each time step—this allows the animat to offer a combination of at least 1 and up to N items, assuming there are N time steps and N categories of items.

In this particular simulation run with 2 categories and 3 subcategories, there were 6 ideas (6 items) to be expressed during the naming game rounds; animats came up with 6 different patterns of messages to express them. Animats did not have to devise complex ways to express these ideas due to the ease of task. It seems that animats assigned each signal to each item, and emitted a signal to each corresponding item.

This simple result may be indicating that languages strive to be as simple as possible. A similar phenomenon in human language is the variety of the number of phonemes that different human languages have: humans have the physiological capability to produce a countless number of sounds, but human languages use only a limited subset of them.

Table 6.2 shows the language used by the top pair at the end of 1200 rounds. Some entries have two values because of the discrepancies between how each of the pair produced and interpreted messages. Once again, the 6 signals are denoted by $\{0,1,2,3,4,5\}$. The 2 categories are denoted by $\{0,1\}$ and the 3 subcategories $\{0,1,2\}$. There are $3^2 = 9$ combinations (ideas) to express.

These animats ended up combining 4 out of 6 available signals ("2," "3," "4" and "5") to express 9 ideas, even though they were only able to distinguish among 8 ideas. The number of ideas expressed is greater than the number of signals the animats are capable of using because animats have evolved to take advantage of sequencing.

By looking at patterns of what signals are used in which messages, I can try to understand if particular signals have a specific meaning. For example, the signal "4" is included at the initial position in all messages communicating the presence of item c0s0 in the request ((0,0), (0,1), (0,2)); therefore, if an animat hears the signal "4" at the head of the message, it might decide that it should offer that item. Similarly, the signal "2" in the initial position seems to specify item c0s1, and "5" item c0s2. However, there are messages that use the signal "5" in the second position that do not have item c0s2, such as (0, 1), whose corresponding message was "45" and (1, 1), described by message "25." It seems that the second signal in each of the messages seems to specify the category-1 item. Take a look at the messages "22," "25" and "23." Even though the signal "5" in the initial position selects item c0s2, "5" in the second position seems to select item c1s1. The signal "3" in the second position, similarly, specifies item c1s2.

It seems that item c1s0 is expressed differently in the two cases that require it: the message for (0, 0) was "44," and the message for (1, 0) was "22." It seems that repeating the first position signal specifies c1s0, but the message "55" was interpreted as (2, 1) instead of (2, 0). This result is interesting; the meaning of a signal depends on its position in the message. In other words, this language that came out of the simulation is more than a signal-meaning association system. Each signal does not simply specify an item, but what it specifies depends on whether it is the first signal of the message or it follows another signal.

			From						
			Input Neuron						Context Neuron
			0	1	2	3	4	5	
To	Hidden Neuron	0	3.375	-2.75	-3.875	-0.0625	-3.3125	0.0625	-3.625
		1	-1.875	-1.625	2.3125	-3.9375	0.125	1.6875	1.5625
		2	-2.5625	-3.5	-0.6875	1.5625	3	-2.625	0.625
		3	-2.375	2.5	-3.5	-2	-1.9375	1.875	-3.25
		4	-3.5625	-2.375	-2.3125	0.3125	1.9375	-3.9375	3.4375
		5	2.25	1.3125	3.6875	0.375	3.125	-0.6875	-3.3125

Table 6.3: Weights on connections into the hidden neurons in the production neural network.

			From					
			Hidden Neuron					
			0	1	2	3	4	5
To	Output Neuron	0	1.5	-1.9375	0.125	2.625	-3.25	-3.9375
		1	-3.125	0.5625	1.5625	-2.0625	-2	-1.5
		2	0.8125	-3.75	-1.4375	3.1875	0.4375	3
		3	-0.1875	3.0625	-0.5625	0.6875	0.75	-0.0625
		4	2.9375	-1.125	-1.625	-3.8125	3.125	3.25
		5	-3.9375	1.25	1.5	-3.25	1.25	3.0625

Table 6.4: Weights on connections into the output neurons in the production neural network.

			From						
			Input Neuron						Context Neuron
			0	1	2	3	4	5	
To	Hidden Neuron	0	-3.8125	1.875	3.125	-2.75	-1.4375	-1.6875	1.5
		1	-2.75	-2	-3.375	-3.4375	-0.3125	1.6875	2.375
		2	1.9375	2.4375	-3	0.875	2.8125	-1.4375	-3.875
		3	2.625	0.1875	-3.9375	3.25	-3.0625	3.5625	2
		4	0.25	0.375	0.5625	-0.0625	1	0	-3.625
		5	0.1875	2.875	-2.4375	2.0625	-3.9375	-2.0625	3.25

Table 6.5: weights on connections into the hidden neurons in the interpretation neural network.

			From					
			Hidden Neuron					
			0	1	2	3	4	5
To	Output Neuron	0	0.875	-0.75	3.4375	-1.1875	0.0625	3.9375
		1	2.4375	-0.5	0	-2.125	1.8125	0.5
		2	-3.375	0.625	-2.75	1.125	2.9375	-0.25
		3	3.8125	1.8125	0.1875	-1.3125	-1.0625	-0.6875
		4	0.6875	3.75	-3.1875	-0.75	-3.4375	2.5
		5	0.25	-1.375	2.125	1.125	-3.75	3.25

Table 6.6: Weights on connections into the output neurons in the interpretation neural network.

Weights of the neural network connections of one of these animats are shown in Tables 6.3 to 6.6. Table 6.3 shows the weights on connections from the input and context layers to the hidden layer in the production network. Table 6.4 describes the weights on connections from the hidden layer to the output layer of the network. Tables 6.5 and 6.6 similarly show weights for the interpretation network.

I have also run cases for $(N, M) = (3, 3)$ case and $(N, M) = (4, 3)$. Figures 6.2 and 6.3, respectively, show their results. There are 400 naming game rounds and 800 combination game rounds. The maximum possible fitness points for each pair in the naming game rounds are 5000 for both. The maximum possible fitness points in the combination game rounds are 15000 in the 3 category case: each trade gives a maximum of (50 $\times$ 3 categories) points, and there are 50 trades, and the fitness points of the two animats are combined, making (50 points $\times$ 3 categories $\times$ 50 trades) $\times$ 2 animats = 15000 animats. Similarly, the maximum possible fitness points in the combination game rounds for the 4 category case is 20000.

The results were not as impressive as the $M = 2$, $N = 3$ case. As the number of categories increase, the animats' ability to associate each item to a signal significantly decreases. The animats are taking longer to associate each item to a

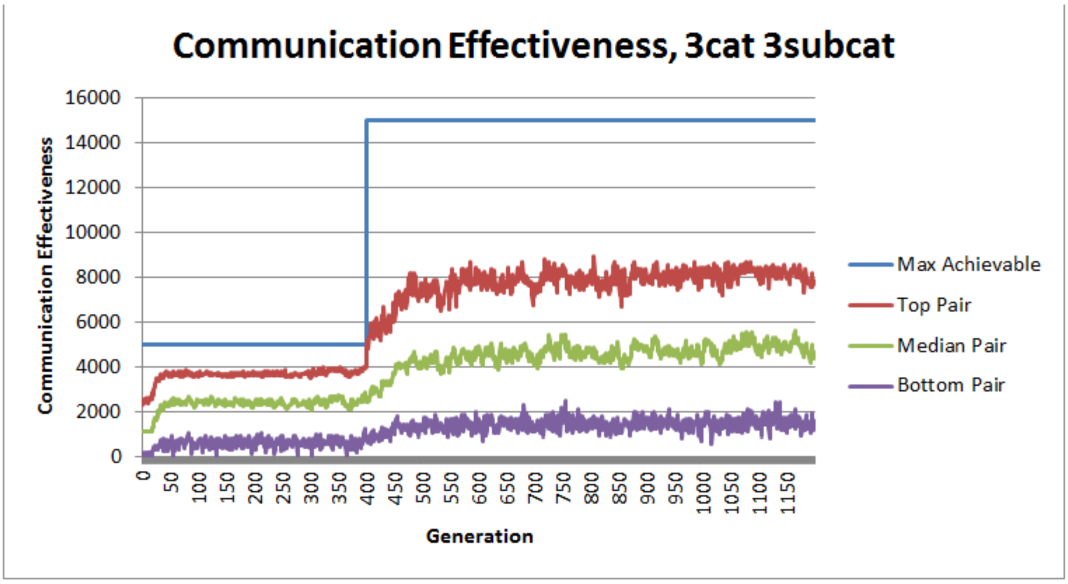

Figure 6.2: Communication effectiveness of animats when there are 3 categories and 3 subcategories of items, with random initial generation. The first 400 generations play the naming game. The latter 800 generations play the combination games.

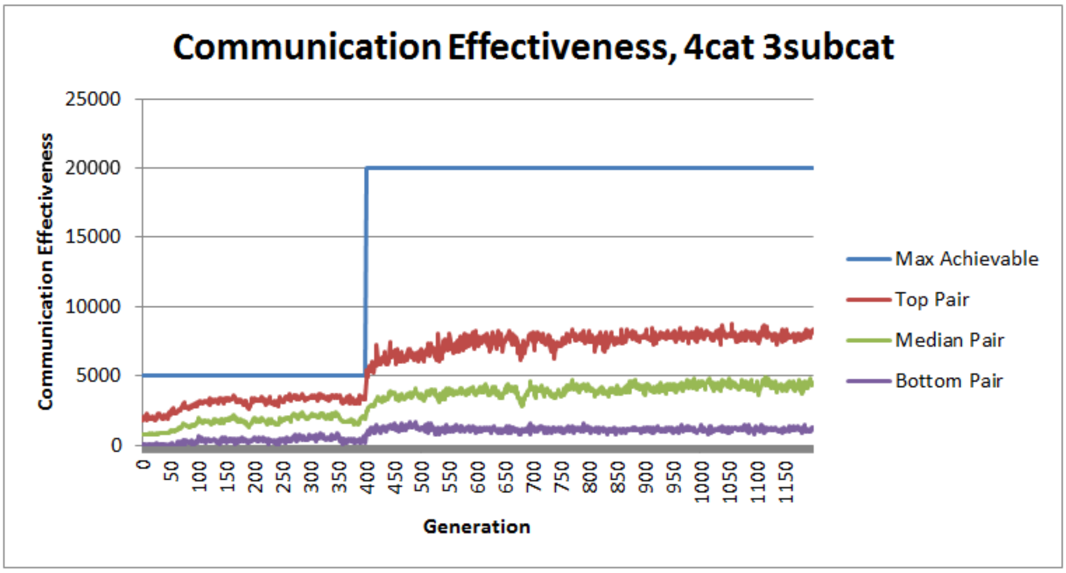

Figure 6.3: Communication effectiveness of animats when there are 4 categories and 3 subcategories of items, with random initial generation. The first 400 generations play the naming game. The latter 800 generations play the combination games.

signal, and they are far from done at the end of 400 rounds. Due to poor performance in these cases, I have decided to conduct further analysis on the 3-category 3-subcategory case in the next section of the experiment, in which the population is assumed to possess animats that play the naming games with perfection. This way, animats can at least associate each item to a signal.

6.2 Seeding

The previous section assumed that the initial generation of animats has random genotypes, and they would have to build word-meaning associations. This part of the experiment assumes that some animats have perfect word-meaning associations established already ("mapped"), and that the population goes directly into combination games. By introducing such individuals to the population, the population as a whole should become capable of associating each item with a word. The question of interest, then, is how the population that is already capable of naming items would adapt to the necessity of expressing multiple ideas in one message.

Figure 6.4 shows the results for simulations runs with 20 "mapped" seeded animats out of 100. There are $N = 2$ categories and $M = 3$ subcategories. The simulation ran 800 rounds of combination games. The maximum possible communication effectiveness (per-pair combined fitness) is 10,000.

The graph shows that there is no significant difference in performance in the 2-category 3-subcategory cases from the unseeded simulation run. Both simulation runs were similar in performance. The differences, then, are most likely due to random variation.

Table 6.7 shows the messages sent to each other at the end of all rounds. The resulting language is similar to the one in the unseeded simulation run.

Another run with seeding was conducted with $N = 3$ categories and $M =$

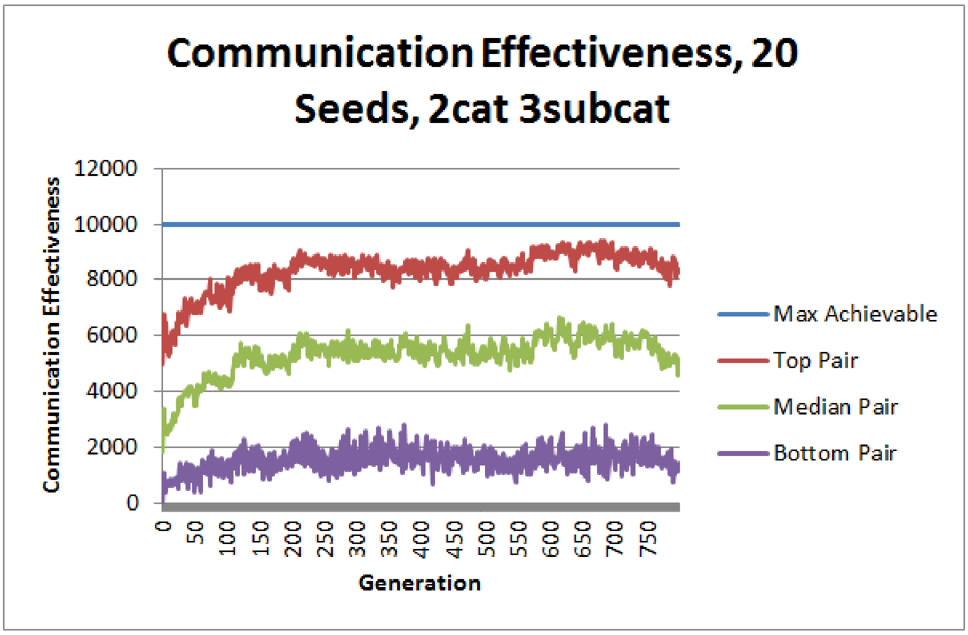

Figure 6.4: Communication effectiveness of animats when there are 2 categories and 3 subcategories of items, with 20 seeded animats. The population plays 800 rounds of the combination games.

Intention	Message	Interpreted as
(0, 0)	30	(0, 0)
(0, 1)	30	(0, 0)
(0, 2)	50	(0, 2)
(1, 0)	35	(1, 0)
(1, 1)	55	(1, 2)
(1, 2)	55	(1, 2)
(2, 0)	33	(2, 0)
(2, 1)	33	(2, 0)
(2, 2)	53	(2, 2)

Table 6.7: Messages sent to each other by the top performing pair at the end of 800 rounds. There are 2 categories and 3 subcategories of items. Communication effectiveness for this top performing pair was 8400 out of 10,000 possible.

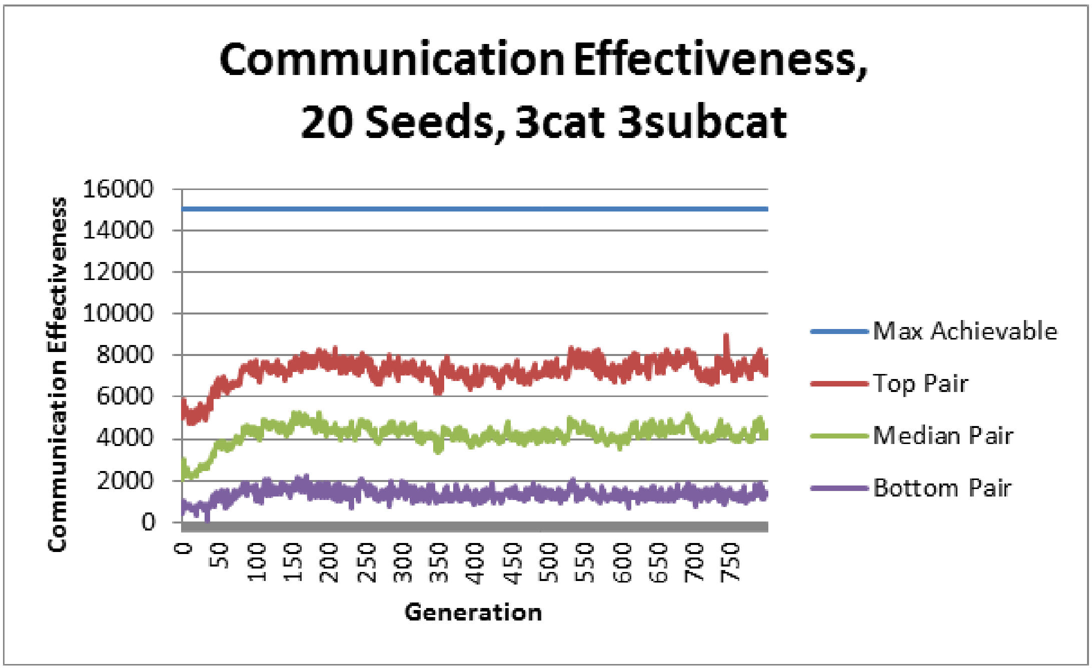

Figure 6.5: Communication effectiveness of animats when there are 3 categories and 3 subcategories of items, with seeding. The population plays 800 rounds of the combination games.

3 subcategories. Recall that, without seeding, the population was not able to efficiently associate each item with a signal, and it was not able to perform well in the combination games. Once again, only 800 combination game rounds and no naming game rounds are conducted. The maximum achievable communication effectiveness value is 15,000 for this case. Figure 6.5 is the graph of communication effectiveness over 800 rounds.

The animats struggled to express $3^3 = 27$ ideas, even with seeding. The possible reason for this may be that this task may require a more complex neural network structure. For example, if the animats were equipped with simple recurrent neural networks with more than one layer of context, then they would be able to "remember" more information, and this may aid them form more message structures. Also, there may be a problem with having way too many ideas to be expressed. With 3 categories and 3 subcategories, there are $3^3 = 27$ combinations to be expressed in 3-signal sequences with 9 types of signals in the vocabulary. This may be a task too complex to complete in only 800 rounds.

Table 6.8 shows how messages were created and interpreted in the final round. There were 9 signals, so the signals are denoted by $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$. Some entries are labeled "nodata." This is due to the fact that trade requests are generated by random.

The table shows a great discrepancy between the partners in both production and interpretation of messages. During the 50 trades, if the animat did not make a certain request, it is labeled "nodata" in the tables. Many of the items were assigned a different message from each animat: to express (2, 2, 1), or, "a combination of items c0s2, c1s2 and c2s1," one animat produced message "000" while the other produced "222." Animats were unable to develop a good way to express these 27 meanings with 9 signals and message length of 3. Instead, animats are seen to limit themselves to a small number of messages in the languages that tries to achieve some amount of performance with small effort. There were two

Intention	Message	Interpreted as
(0, 0, 0)	777 / 222	(1, 0, 1) / (2, 2, 0)
(0, 0, 1)	777	(1, 0, 1)
(0, 0, 2)	nodata / 222	nodata / (2, 2, 0)
(0, 1, 0)	777 / 222	(1, 0, 1) / (2, 2, 0)
(0, 1, 1)	777	(1, 0, 1)
(0, 1, 2)	777 / 211	(1, 0, 1) / (2, 0, -)
(0, 2, 0)	000 / nodata	(1, 2, -) / nodata
(0, 2, 1)	777 / 222	(1, 0, 1) / (2, 2, 0)
(0, 2, 2)	777 / 222	(1, 0, 1) / (2, 2, 0)

Intention	Message	Interpreted as
(1, 0, 0)	nodata / 722	nodata / (1, 0, 0)
(1, 0, 1)	777	(1, 0, 1)
(1, 0, 2)	777	(1, 0, 1)
(1, 1, 0)	nodata / 122	nodata / (-, 0, 0)
(1, 1, 1)	777	(1, 0, 1)
(1, 1, 2)	777 / 111	(1, 0, 1) / (-, 0, -)
(1, 2, 0)	222 / 222	(2, 2, 2) / (2, 2, 0)
(1, 2, 1)	777	(1, 0, 1)
(1, 2, 2)	000 / nodata	(1, 2, -) / nodata

Intention	Message	Interpreted as
(2, 0, 0)	222 / nodata	(2, 2, 2) / nodata
(2, 0, 1)	777 / nodata	(1, 0, 1) / nodata
(2, 0, 2)	nodata / 222	nodata / (2, 2, 0)
(2, 1, 0)	nodata / 222	nodata / (2, 2, 0)
(2, 1, 1)	277 / 222	(2, 0, 1) / (2, 2, 0)
(2, 1, 2)	200 / nodata	(2, 2, 1) / nodata
(2, 2, 0)	222 / 222	(2, 2, 2) / (2, 2, 0)
(2, 2, 1)	000 / 222	(1, 2, -) / (2, 2, 1)
(2, 2, 2)	000 / 222	(1, 2, -) / (2, 2, 1)

Table 6.8: Messages sent to each other by the top performing pair after 800 combination game rounds. There are 3 categories and 3 subcategories of items.

messages, “777” and “222,” that were frequently used by both of the animats, but their usage was not agreed upon, since both the situation at which they were used and how they were interpreted differ between the animats. It seems that animats were still in the process of agreeing upon a language that works well when the 800 rounds finished.

This disappointing outcome of the seeded experiment suggests that, even if signal-to-meaning associations are perfect in the beginning, they are not helpful in obtaining maximal performance under the circumstances. The biggest problem of this simulation run is that there are so many combinations to distinguish from. The graph did not show much improvement during the 800 rounds; the simulation may never have reached an optimal performance point even if it was run for a much longer time. Simple recurrent neural networks with one context layer were not able to take on this task effectively; neural networks of more complexity may be able to handle the problems.

6.3 Learning

Simulations with imitation learning include a parent-to-child imitation learning stage during the life of each animat. As explained in one of the previous sections, this imitation learning scheme works by feeding the parent and child with the same input and comparing their outputs. The discrepancies are used to update the child’s neural network weights using backpropagation. The parent and child go through this process for 200 examples. Learning production feeds trade requests and compares the produced signals. Learning interpretation feeds parent signals and compares the items offered.

Figure 6.6 shows the communication effectiveness graph for such a run with 2 item categories and 3 subcategories. There are 400 naming rounds and 800 combination game rounds.

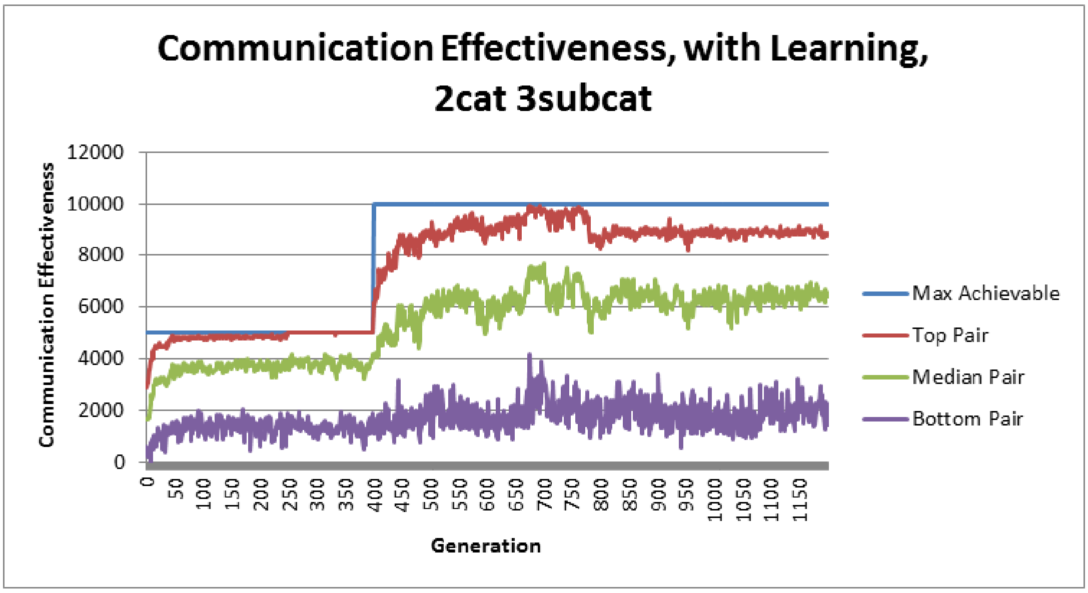

Figure 6.6: Communication effectiveness of animats when there are 2 categories and 3 subcategories of items, with all animats learning by imitating parents at birth. The first 400 generations play the naming game. The latter 800 generations play the combination games.

The results are significantly better than the case without learning. The population developed an effective language much more quickly. This is most probably because learning from parents makes it easier for the population to agree upon the use of signals and messages. The result shows the role of learning in the development of language: learning helps a population come into consensus of what the language should look like.

Simulation runs with 3 item categories or more were also conducted but were equally unsuccessful as the simulation runs without learning. For this reason, the data for these runs are not included in this paper.

CHAPTER 7

Discussions

7.1 Analysis

It is apparent from the results that complex language requires complex neural structure. The neural network structure I used—simple recurrent neural networks with one context layer—did allow messages that combined different signals to convey different meanings. However, the complexity of how these messages were constructed was limited. In particular, they performed poorly on 3-category cases.

The biggest problem I faced in the project was that small changes in the complexity of the environment greatly increase the difficulty of the animats' tasks. With 3 subcategories of items, increasing the number of categories from 2 to 3 meant that the total number of combinations of items increased from 9 to 27. Then, even if the animats have the capability to handle this complexity, they would take a long time evolving their neural networks to handle them with an effective language. If it takes a long time, simulation runs will start to require days to complete. This scaling issue will continue to threaten the project even if optimizations are made.

To address these issues, I could make the neural network more complex. For example, I could increase the number of hidden neurons and context neurons to see if they help the animats solve the problem better. Having more than one layer of context will allow animats to have longer short term memory; they will not only remember the previous signal but also several previous signals. This will allow animats to form more complex messages. However, this adds a new problem of search space scalability: because there will be so many connections and their weights, there will be a much bigger number of possible configurations of the network, and this will most likely make it hard for the simulation to find an optimal configuration.

In all, my experiment demonstrates some success in modeling signal sequence language communication. Human language is a sequence of words, or sequence of phonemes. The parallels between the sequential nature of human language and the signal sequences in the project are apparent. The simulation has succeeded in having animats develop communication to take advantage of sequencing and combination with a simple neural structure. Also, by comparing the simulation runs with and without imitation learning, I was able to demonstrate the role of learning in the development of language. On the other hand, the results pointed to difficulties faced by the models design, and there are many simplifications in the model that should ideally be made more realistic. The project leaves several issues yet to be addressed.

7.2 Possible Future Extensions

Though the methodology is popular, simulating language development primarily using evolution may be a bit unrealistic, because language as we use it seem to be more cultural, though it is the genetics and physiology that makes it possible. In order to focus on the topics I was most interested in, I have made my model to be relatively simple, at the cost of realism. The next step for this project, then, is to steer towards realism.

There are many ways that the project can become more realistic and natural. One way is to expand the range of actions that animats can take. This would make the environment more complex, perhaps adding the concepts of space, distance and direction to the simulation. In this case, animats would move around and only communicate locally, and communication would affect the actions they would take. By allowing more variety of action, I could expect to see more complexity in language. This would also require more complex neural network structures to allow animats to make more complex messages.

Another simplification that should ideally be redone in the future is the population dynamics. In the natural world, generations of animats are not clear-cut, and selectional pressure should be fitness instead of communication effectiveness. By making communication effectiveness to be a direct selectional pressure, I was able to make the population of animats develop language, but communication effectiveness should ideally be only an indirect selectional pressure. Instead, communication effectiveness should help animats gain fitness, and fitness will in turn increase the animat's reproductive chances.

One feature I would like to enhance in the project is learning. After all, language is learned and culturally transmitted. Communication itself has an effect on the understanding of the language. In the future, I would like to make this language development process more dynamic by allowing animats to learn not only by imitating parents but also from each other. In this case, each conversation that an animat takes part in becomes a learning experience that affects its understanding of the language. This is much more realistic, as humans also deepen their understanding of language by conversing with others. Humans also learn by other methods, such as reading about the language's grammar or acquiring vocabulary by reading a dictionary. In the end, we will end up with a simulation model that aims to mimic humans' ability to learn and use a language as closely as possible.

References

[AT96] T. Arita and C. E. Taylor. "A Simple Model for the Evolution of Communication." *The Fifth Annual Conference on Evolutionary Programming*, **V**:405–409, 1996.

[Can01] A. Cangelosi. "Evolution of Communication and Language Using Signals, Symbols, and Words." *IEEE Transactions on Evolutionary Computation*, **5**(2):93–101, 2001.

[Smi02] K. Smith. "The cultural evolution of communication in a population of neural networks." *Connection Science*, **14**(1):65–84, 2002.

[Ste05] L. Steels. "The emergence and evolution of linguistic structure: from lexical to grammatical communication systems." *Connection Science*, **17**(3–4):213–230, 2005.

[Wer90] Paul J. Werbos. "Backpropagation Through Time: What It Does and How to Do It." *Proceedings of the IEEE*, **78**(10):1550–1560, 1990.