Zébulon Goriely

Simulating Language Learning and Evolution

Computer Science Tripos – Part II

Queens' College

January 21, 2020

Proforma

Name:	**Zébulon Goriely**
College:	**Queens' College**
Project Title:	**Simulating Language Learning and Evolution**
Examination:	**Computer Science Tripos – Part II, May 2020**
Word Count:	**NULL**[1]
Project Originator:	Zébulon Goriely
Supervisor:	Prof. Paula Buttery and Dr. Andrew Caines

Original Aims of the Project

- max 100 words
- Investigate evolutionary advantage of language
- Do this by implementing a simulation of a population of neural nets in a toy mushroom world
- Compare the fitness of three species; one without language, one with an evolved language and one with an externally imposed language

Language has evolved and therefore probably gave an evolutionary advantage to the individuals that exhibited it. According to Cangelosi and Parisi (1998), it is difficult to investigate the evolutionary origin of language and the selective pressures that may have originated language due to the limited evidence available. They propose using computer simulations of evolutionary scenarios to investigate this. In the paper referenced, they describe a simulated toy world where agents controlled by neural networks interact with an environment of mushrooms that are edible and poisonous. Exploring this simulation is the basis of my project.

Work Completed

- max 100 words
- Implemented the simulation capable of hosting the three population types
- Genetic algorithm, feed-forward neural networks
- Graphing etc.

[1] This word count was computed by `detex diss.tex | tr -cd '0-9A-Za-z \n' | wc -w`

I implemented the mushroom-world simulation and explored three populations of feed-forward neural networks; one without language, one with an externally imposed language and one with an evolved language. To simulate evolution, I implemented a genetic algorithm to allow the fittest members of each generation to reproduce. The state of the simulation was saved at each generation in order plot the fitness of the three populations, plot the quality of the language produced and investigate behavioural tests. To conclude the project, I compared these findings with Cangelosi and Parisi (1998).

Special Difficulties

None.

Declaration

I, Zébulon Goriely of Queens' College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed Zébulon Goriely

Date [date]

Contents

List of Figures

Acknowledgements

Paula et al.

Chapter 1

Introduction

Introduction here, start again as if abstract does not exist. Include your contributions in bullet point format. End with paragraph describing structure of the rest of the paper

1.1 Motivations

High level, still cite, why does this interest us from a scientific perspective. What could we learn.
- Want to investigate evolutionary origin of language
- Difficult to investigate due to limited evidence, so computer simulations could be good
- Investigation of cultural vs sexual mechanisms
- Artificial life
- Main question - how can language evolve when it has a purely informative function and so advantageous to receiver but not to the producer?
- Reference Clark (1993) for what an efficient language is

1.2 Prior Work

State what similar work has been done before and since what i'm trying to replicate. Also a continuation of the motivation. Why does this paper in particular interest you. Focus more on their prior work. Talk about work that has continued.
- Symbol grounding
- Language games
- Mushroom world

1.3 Project Overview

Chapter 2

Preparation

2.1 Mushroom World

2.1.1 Mushrooms

- Talk about how mushrooms are represented
- Differences between edible and poisonous mushrooms

2.1.2 Environment

- Talk about size of the world, number of mushrooms in the world
- Number of epochs, cycles

2.2 Entities

- Entities represent animals that eat edible mushrooms

2.2.1 Feed-Forward Neural Networks

- Description of what a neural network is
- Description of feeding forward
- Description of structure used for these entities

2.2.2 Genetic Algorithm

- Description of algorithm
- Specific values used for this simulation
- Represents evolution, natural selection, etc.

2.2.3 Population types

- No language
- External language
- Evolved language

2.3 Evaluation Metrics

2.3.1 Generational Fitness

- 'Fitness' described as average score of all entities for each generation where score is calculated using total number of edible and poisonous mushrooms eaten by the entity
- Plot fitness across 1000 generations for each of the three population types

2.3.2 Efficiency of Language

- 'Naming experiment' to graph the language used for edible and poisonous mushrooms
- Describe how QI encapsulates the 'efficiency' of a language
- Talk about how we can investigate how language production is linked to cognitive ability to differentiate between mushrooms by looking at language produced by non-speaking populations
- Should measure correlation between QI and fitness

2.4 Requirements Analysis

My project reimplements Cangelosi and Parisi (1998) soooo As such, the requirements of this project are:
- Implement simulation environment (mushrooms, cycles,)
- Implement neural-network entities with genetic algorithm, three populations
- Plot fitness over time
- Plot distributions of signals used for edible and poisonous mushrooms
- Calculate QI over time and investigate correlation with fitness

2.5 Starting Point

- coded from scratch
- familiar with neural networks from AI course, language theory from Formal Models of Language
- familiarised self with genetic algorithm

2.6 Software Engineering

2.6.1 Languages and Libraries

python, numpy, pytest

2.6.2 Project Management

- Agile
- How I stuck to project plan

2.6.3 Version Control

- github, hardware backups

2.6.4 Development tools

travis, pylint, yapf

2.6.5 Development environment

VSCode, Slurm w/ HPC

2.6.6 Code License

MIT?

2.7 Summary

Chapter 3

Implementation

Start with UML, high level overview
Talk about problems with trying to replicate, not knowing what you set.

Chapter 4

Evaluation

Show graphs, run by run etc.
Variance is interesting to discuss
Could also be interesting to query what happens in an elbow in the graph - visual and a bit subjective. Three categories - those that don't take off, those that suddenly take off and those that gradually take off.
Variance in each run, consequences of initialisation/activation, bigger questions about how reliable neural networks are, whether they're a good model for language evolution.

Chapter 5

Conclusion

Bibliography

Cangelosi, A. and Parisi, D. (1998). The emergence of a 'language' in an evolving population of neural networks. *Connection Science*, 10(2):83–97.

Appendix A

Project Proposal

Computer Science Tripos – Part II – Project Proposal

Simulating Language Learning and Evolution

Zébulon Goriely — Queens' — zg258
Originator: Zébulon Goriely
18 October 2019

Project Supervisor: Prof. Paula Buttery and Dr. Andrew Caines
Director of Studies: Prof. Alastair Beresford
Project Overseers: Prof. Pietro Lio & Dr. Robert Mullins

Introduction

Language has evolved and therefore probably gave an evolutionary advantage to the individuals that exhibited it. As Angelo Cangelosi and Domenico Parisi described in a 1998 paper[1], it is difficult to investigate the evolutionary origin of language and the selective pressures that may have originated language due to the limited evidence available. They propose using computer simulations of evolutionary scenarios to investigate this. In the paper referenced, they describe a simulated toy world where agents controlled by neural-networks interact with an environment of mushrooms that are edible and poisonous. This simulation and the ideas explored in the paper will be the basis for my project.
In the paper, Cangelosi and Parisi use small feed-forward neural networks to control the behaviour of each agent. The weights are initially random; a genetic algorithm is used to improve the fitness of the species over many generations. The agents are also given linguistic abilities; input and output nodes of the neural networks produce signals that allow for communication.

[1] `https://doi.org/10.1080/095400998116512`

By creating three different populations (one without language, one with an externally imposed language and one with an evolved language) we can investigate the evolutionary advantage of language. Furthermore, it allows us to investigate a key question posed in the paper: *"Since language requires the parallel evolution of linguistic production and linguistic comprehension, how can language evolve when it has a purely informative function and therefore it is advantageous to the receiver but not the producer?"*
For this project, I will re-implement the simulation described. I will then create analysis tools to investigate the findings of the paper to see if I observe the same results.

Starting Point

I have a small amount of experience in programming simulations; for my A-Level project in 2016, I created a simulation of virus propogation between mosquito and human agents in the Unity game engine.
I do not have any experience programming neural networks, however, I am confident that I understand the backpropagation algorithm and basic neural network structure through the Artificial Intelligence course I took last year. In the papers I plan to reference, Cangelosi very clearly describes the structure of the neural networks he uses and I am confident that I will be able to follow his work.
Over the summer I read a book titled Simulating the Evolution of Language which gave me an overview of the techniques used in this field. Alongside the Formal Models of Language course that I took last year, I now have a sufficient base of understanding to begin this project.

Work to be Done

The work for this project can be roughly divided into two stages; implementing the simulation and constructing the means of evaluating my implementation against the findings in the original paper. I will also regularly be creating tests to evaluate my simulation.

Implementing the Simulation

1. Set up the simulation environment by creating the world grid and implementing the properties of poisonous and edible mushrooms. Create the simulation loop divided into regular 'epochs'.

2. Create the agents for the simulation, giving them position and energy properties.

3. Implement feedforward neural networks to control the behaviour of the agents; input units to identify the location of the nearest mushroom, visual perception units to observe mushroom properties (only when close enough) and signal perception units for when language is implemented. The output units control the movement of the agent and production of signals. There will also be hidden units.

4. Implement the genetic algorithm that runs after all agents complete the simulation. The fittest agents are determined by the energy level (based on eating edible mushrooms and avoiding poisonous mushrooms). The fittest agents are then chosen for asexual reproduction, producing offspring that have genetic mutations in the form selecting a percentage of the weights to change by a random amount.

5. Create two different populations, one without language (where the signal perception units are always set to the same, constant value) and one with an externally imposed language (where the signal perception units are set to one of two signals depending on the type of the nearest mushroom).

6. Create a third population with an evolved language. Instead of externally imposed signals, in each simulation cycle one of the other agents is randomly selected as the 'speaker' and its output is connected to the input signal perception units of the 'listener'.

Analysis

1. Plot the average fitness over the number of generations to compare between the three populations.

2. Produce some behavioural tests to investigate the behaviour of random individual organisms at specific generations.

3. Plot the frequency distribution of the different signals produced by the individuals with the evolved language using a 'naming task'.

4. Calculate the Quality Index (QI) of the language produced by the population without language and the population with an evolved language to investigate the genetic advantage of producing productive signals. The QI evaluates the efficiency of a language based off of three criteria; (1) functionally distinct categories are labeled with distinct signals, (2) a single signal tends to be used to label all the instances within a category and (3) all the individuals in the population tend to use the same signal to label the same category.

5. Investigate the correlation between QI of the language and the fitness of the species to determine if change in the language or in the categorisation skill of the agents affects the other ability.

Testing

To evaluate my project and ensure that my simulation implementation is functional, I will create an ensemble of unit tests for each of the tasks above. These will be created in parallel as I develop each part of the implementation. For the simulation, this will involve small examples or scenarios to show that each part of the simulation is fully functional.

Success Criterion

The project will be deemed a success if I can implement the simulation as described in the tasks above (evaluated by my unit tests) and if I can implement the analysis tools to compare the findings of my implementation to the findings of the original paper.

Timetable and Milestones

I've broken down this timetable into two and three week intervals. At the end of December, I will be writing my Progress Report and simultaneously making adjustments to the timetable as needed.

25th October – 10th November

Middle of Michaelmas Term. Includes first deadline for NLP coursework.
Task: Create a high-level design of the system. Set up the project files with a version-control system. Experiment with creating small simulations in python and do suitable research into Neural Network libraries.
Milestones: Have a git repository with project files. Have a design plan with specific details of the simulation confirmed.

11th November – 24th November

Middle of Michaelmas Term.
Task: Complete implementation tasks 1 and 2 as described above. Experiment with adding Neural Networks to control the behaviour of the agents.
Milestones: Have a working simulation environment with poisonous and edible mushrooms. Have agents with positions and energy values but no functional neural networks yet.

25th November – 8th December

End of Michaelmas Term. Includes second and third deadline for NLP coursework.
Task: Complete implementation tasks (3). Start working on implementation task (4).
Milestones: Have the agents successfully controlled by neural networks. Be able to run an entire lifespan of one agent within the simulated world.

9th December – 22nd December

Christmas holiday. Will likely be in Cambridge to help with Queens' interviews.
Task: Complete implementation tasks (4), (5) and (6). Also aim to complete analysis tasks (1) and (3).
Milestones: Have a fully functional simulation that allows for running a thousand generations of a population of agents. Have three different populations to compare; one without language, one with an externally imposed language and one with an evolved language.

Have a tool to graph the average fitness of each population over the number of generations and another tool to view the probability distribution of the signals chosen for the evolved language over the number of generations.

23rd December – 12th January

Christmas holiday. Will take a break to revise Michaelmas courses and to spend time with family.
Task: Complete analysis task (2). Write the Preparation chapter of the Dissertation. Review the timetable for the remainder of the project and adjust in light of experience so far. If ahead of schedule, plan time for extensions. Start to plan tests cases.
Milestones: An outline of the dissertation document with a completed Preparation section.

13th January – 2nd February

Start of Lent term. Will have regular labs for Mobile Robot Systems.
Progress Report Deadline: 31st January
Task: Write the Progress Report. Start to fill out the Implementation chapter of the Dissertation. Complete analysis tasks (4) and (5).
Milestones: Progress report submitted and entire project reviewed both personally and with overseers. Have tools to plot the Quality Index of the evolved language against the fitness of the population. At this point, all the tasks in the **Work to Do** section will have been completed, satisfying the **Success Criteria**.

3rd February – 23rd February

Middle of Lent term. Both deadlines for the Mobile Robot Systems assignments.
Task: Begin analysis of the simulation. Begin to work on extensions to the project, keeping in mind time needed to write the Dissertation.
Milestones: Have the start of a test suite with a series of diagrams to use to evaluate my implementation.

24th February – 15th March

End of Lent term. Deadline for the Mobile Robot Systems mini-project report.
Task: Complete testing. Evaluate the outcomes of the tests against the findings in the original paper. At this point, the second half of the **Success Criteria** will have been achieved. If needed, revise the implementation to be clean, documented and consise. Work on other extensions.
Milestones: Examples and test cases run with results collected. Code should perform a variety of interesting tasks and should be in a state that in the worst case it would satisfy the examiners with at most cosmetic adjustment

16th March – 5th April

Start of Easter holiday. Might stay in Cambridge for part of it to work. Will balance revision and work on the project.
Task: Complete work on any extensions. Draft the Evaluations and Conclusions chapters of the Dissertation.
Milestones: Extensions almost complete. Skeleton of entire Dissertation in place.

6th April – 19th April

End of Easter holiday. Might get back to Cambridge early to work. Will balance revision and work on the project.
Task: Complete the Implementation and Introduction chapters of the Dissertation. Send the full draft to Director of Studies and Supervisors by 21st of April.
Milestones: Dissertation essentially complete, with large sections of it proof-read by Supervisors and possibly friends and/or Director of Studies.

20th April – 8th May

Start of Easter Term. Will be balancing revision, lectures and final work on the project.
Final Deadline: 8th May
Task: Finish Dissertation, preparing diagrams for insertion. Review the whole project, checking the Dissertation and spending the final few days on whatever is in greatest need of attention. Aim to submit the dissertation at least a week before the deadline.
Milestone: Submission of Dissertation

Possible Extensions

Graphic Visualisation

As a side extension, I could implement a visual interface to watch the life of one agent within the simulation. This would involve rendering a simple 2D world with textures for the agents and mushrooms. This could be expanded further by adding a User Interface for setting up the simulation and having windows showing the progress as it occurs live.

Symbolic Theft vs. Sensorimotor Toil

In a 2000 paper[2], Cangelosi and Harnad use a similar same toy world of mushrooms and foragers to place two ways of acquiring categories in direct competition with each other. They compare "sensorimotor toil" (where categories are acquired through real-time, feedback-correct, trial and error experience) to "symbolic theft" (where new categories are acquired by hearsay from boolean combinations of symbols *describing* them). They find

[2] `http://cogprints.org/2036/`

that the origins of natural language could be explained by the apparent infinitely superiority of a hybrid symbolic/sensorimotor combination compared to purely sensorimotor precursors.
As an extension, I could expand my simulation to investigate the findings of this paper. This involves implementing a more complicated neural network, adding supervised learning through back-propagation, implementing more sophisticated mushroom features and expanding the simulation to host multiple populations at once.

Investigating the Evolution of Syntax

In a 1999 paper[3], Cangelosi expands the toy mushroom world simulation further to investigate how languages that use combinations of words (such as the "verb-object" rule) can emerge by auto-organisation and cultural transmission. Mushrooms are either edible or poisonous but also have one of three colours - the edible mushrooms of a particular colour correspond to a particular action in response.
In this extended simulation, after the first 300 generations parents and children co-exist within the simulated world. Parents teach the evolved language to their children. Children undergo a Listening Task (where parents describe the closest mushroom) and a Naming Task (where the mushroom name is used for supervised learning through backpropagation).
This is a substantial increase in complexity but allows would allow me to investigate the evolution of a more complex language.

Resources Declaration

For this project, I plan to use my computer, (2.8 GHz CPU, 16 GB RAM, 750GB Flash Storage, macOS Mojave). The code will be regularly pushed to a GitHub repository to be able to recover from failure or loss on my local machine. I will also create weekly backups on an external hard-drive to provide another source of recovery. Should my machine fail, I will be able to continue working on an MCS machine. I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure.
I will also need to use the high powered computer when running large simulations to save processing time.

[3] `https://link.springer.com/chapter/10.1007/3-540-48304-7_86`