Simulating Biological Evolution as a Learning Algorithm: An Empirical Study with Multiple Target Concepts

Nico Fidalgo & Puyuan Ye

CS 2280

March 31, 2025

1 Introduction

Biological evolution is increasingly interpreted as a learning process in which populations adapt over successive generations through mechanisms that resemble algorithmic learning. The concept of *evolvability*, as introduced by Valiant, along with foundational work by Holland and Goldberg, provides a framework for understanding how complex functions can be learned via evolutionary processes. In this project, we propose to simulate a genetic algorithm (GA) that evolves candidate solutions to approximate various Boolean target functions. By exploring multiple target concepts, we aim to add complexity to the project and investigate how different target functions influence convergence behavior and learning efficiency.

2 Research Questions

2.1 Objective

Develop and implement a GA-based simulation that learns several target Boolean functions by evolving candidate solutions.

2.2 Research Questions

1. How do evolutionary parameters (mutation rate, population size, and selection pressure) affect the convergence speed and accuracy for different target concepts?
2. Which target functions (e.g., simple conjunctions, disjunctions, parity, and majority functions) present greater challenges for the GA, and how does performance vary across these functions?
3. Can we identify parameter regimes that consistently yield high fitness across diverse target functions?

3 Methodology

3.1 Labeled Data Generation

- **Input Data:** We will generate synthetic datasets consisting of binary vectors (e.g., 4-bit vectors).

- **Target Concepts:** For each dataset, labels are assigned using a specific Boolean target function. The concepts we plan to investigate include:
  1. **Simple Conjunction:** The output is 1 if all selected bits (e.g., the first two bits) are 1.
  2. **Simple Disjunction:** The output is 1 if at least one of the selected bits is 1.
  3. **Parity Function:** The output is 1 if an odd number of bits are 1.
  4. **Majority Function:** The output is 1 if the majority of the bits are 1.
- For each input vector x, the corresponding label y is computed as $y = f(x)$ where f is the chosen target function.

3.2 Genetic Algorithm Framework

3.2.1 Representation

Each candidate solution is encoded as a binary string representing a hypothesis for the target function. The candidate's encoding determines how it processes input vectors to generate predictions.

3.2.2 Fitness Function

The fitness function quantitatively evaluates how well a candidate approximates the target function. For a candidate solution:

$$\text{Fitness(candidate)} = \frac{\text{Number of correct predictions}}{\text{Total number of examples}}$$

- **Prediction:** A candidate's hypothesis is applied to each input x via a prediction function.
- **Comparison:** The candidate's prediction is compared with the ground truth label obtained by applying the target function $f(x)$.
- **Scoring:** The number of matches is normalized by the dataset size, yielding a fitness score between 0 and 1.

3.2.3 GA Operations

1. **Initialization:** Randomly generate an initial population of candidate solutions.
2. **Selection:** Use a selection method (e.g., tournament selection) to favor candidates with higher fitness.
3. **Variation:** Apply mutation (bit-flip with a small probability) and optionally crossover (e.g., one-point crossover) to produce new candidates.
4. **Replacement:** Form a new generation by replacing less-fit individuals with the offspring.

The GA iteratively updates the population, and over successive generations, candidates with higher prediction accuracy propagate their genetic material, leading to convergence toward an accurate approximation of the target function.

3.3 Experimental Design & Analysis

We will conduct systematic experiments by varying key parameters such as mutation rate, population size, and selection pressure. For each target concept, we will record metrics including the best and average fitness per generation, convergence speed, and final test accuracy. Comparative analysis across different target functions will help determine which concepts are more challenging and which parameter regimes yield optimal performance.

4 Expected Outcomes

We expect the GA to converge toward candidate solutions with high accuracy for each target concept under appropriate parameter settings. The experiments should reveal how the complexity of the target function (e.g., the difference between a simple conjunction and a parity function) influences the evolutionary process. Moreover, identifying optimal parameter regimes across multiple concepts will provide insights into the relationship between evolutionary dynamics and learning theory.

5 References

Key references for this project include (for now):

- Valiant, L. (2009). *Evolvability.*
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.*
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.*
- Kearns, M. & Valiant, L. (1989). *Cryptographic Limitations on Learning.*