Genetic algorithms snake game

Jeffrey Roed, Rasmus Kibshede, Joachim Richter

May 28, 2024

Abstract

TBA

1 Introduction

The primary objective of this study is to demonstrate how genetic algorithms (GAs) can be utilized to develop an agent that plays the Snake game. The Snake game, a classic arcade game, involves navigating a snake to collect food items while avoiding collisions with walls and the snake's own body. The GA is used to optimize the snake's behavior over successive generations, improving its ability to survive and collect food.

1.1 Research question

How can we develop a genetic algorithm capable of outperforming the average score achieved by our group in the game of Snake?

2 Method

2.1 Selection

The selection methods is responsible for choosing parent chromosomes that will generate the next generation of individuals. Below are some thoughts we have made in choosing the right selection methods for our Snake game.

- **Roulette Wheel Selection.** Individuals are chosen with a probability proportional to their fitness. This maintains genetic diversity and prevents premature convergence [**goldberg1991comparative**].

  Using this method early in the training process is advantageous when the fitness landscape is smooth and diverse strategies need to be explored.

- **Tournament Selection.** Randomly selected individuals compete, and the winner is chosen for breeding.

  Will be useful when the algorithm progresses and a more refined selection is needed to accelerate the optimization process [**goldberg1991comparative**].

- **Rank Selection.** Individuals are ranked based on fitness values, and selection probabilities are based on ranks. This reduces the risk of premature convergence.

  Will be useful throughout the evolutionary process by providing steady selection pressure and maintaining diversity [**srinvas1994adaptive**].

To determine the most effective selection method for our GA, we propose to evaluate each approach by conducting multiple runs of the Snake game simulation. The performance will be assessed based on the average score achieved by the evolved agents over a fixed number of generations. This will help us determine the most suitable strategy for our Snake game agent. Adaptive approaches to selection, as discussed by Srinivas and Patnaik (1994), may also be considered to enhance the overall performance and robustness of the algorithm [**srinvas1994adaptive**].

2.2 Fitness

The fitness function evaluates how well each individual in a population solves the given problem, assigning a numerical score based on their performance. This score guides the selection process for reproduction, favoring higher-scoring individuals to pass their traits to the next generation. By driving the evolutionary progress and maintaining diversity, the fitness function helps the algorithm converge toward optimal or near-optimal solutions

2.3 Crossover

The crossover methods are used in GAs to combine the genetic information of two parent chromosomes in order to generate new offspring. Below are some thoughts we have made in choosing the right crossover methods for our Snake game.

- **Single point Crossover** A random crossover point is selected on the parent chromosomes. All of the genetic information is copied from one parent, and all the genetic information after this point is copied from the other parent. This is a simple method that keeps the genetic information of both parents making is useful for upholding diversity in the population.

  This will be beneficial in the early stages of training, when we need to explore diverse strategies.

- **Two-Point Crossover** Selects two random crossover point on the parent chromosomes. The genetic information is interchanged between the parents to create two new offspring. This method provides a higher level of mixing compared to single-point crossover, which can lead to more varied breeding.

  This will be useful in later stages of training, where we need to fine tune the algorithm strategies by combining the most desirable features of parents.

As highlighted by Henrik Strøm [**strom2024travelling**], the biggest challenge in implementing crossover for problems such as Traveling Salesman Problem (TSP), is building an effective crossover mechanism. TSP is about finding the shortest possible route to visit a number of cities only once,and afterwards return to the starting city. Similar considerations apply to our Snake game GA, where the crossover method must effectively combine parent strategies without breaking the underlying structure of the Nerual Network (NN).

In order to address these challenges, we will also explore crossover methods designed for TSP, such as; Order Crossover, and Partially Mapped Crossover, which maintains validity of the offspring. We will during our research experiment with different techniques in order to find the most optimal crossover method.

2.4 Mutation

The mutation function aims to modify the genome of the offspring. These modifications serve to maintain and introduce diversity in the genetic population, which is crucial for the overall success of the genetic algorithm, and prevents premature convergence.

> *"The role of mutation in GAs has been that of restoring lost or unexplored genetic material into the population to prevent the premature convergence of the GA to suboptimal solutions."* [**srinvas1994adaptive**]

3 Analysis

3.1 Selection

3.2 Fitness

3.3 Crossover

3.4 Mutation

4 Findings

4.1 Selection Enhancement

4.2 Crossover Enhancement

4.3 Mutation Enhancement