Your very own Evolutionary Loop

Natural language and Artificial Intelligence Group

Last Time

- You built an interpreter for a programming language that was always syntactically valid
- You created functions that import user input
- You created ways or generating and running random programs

Now You Need Evolution

Evolutionary Loop

- Generate a population of random individuals
- Gauge their fitness based on some fitness function
- Create a new generation of individuals by breeding the best from the previous generation
- Repeat

Next Steps

- Create "individuals" that contain fitness values and random programs
- Create a fitness function that evaluates the individuals program
- Create a selection method to find parents
- Create mutation and crossover

Individuals

- Create objects that have a float that represents their fitness and a reference to a push program

Fitness Function

- Create a fitness function that will take in x data points and execute the individual's program, replacing all instances of x with the current data point
- Calculate the distance from the expected to the actual
- Average that and call it fitness

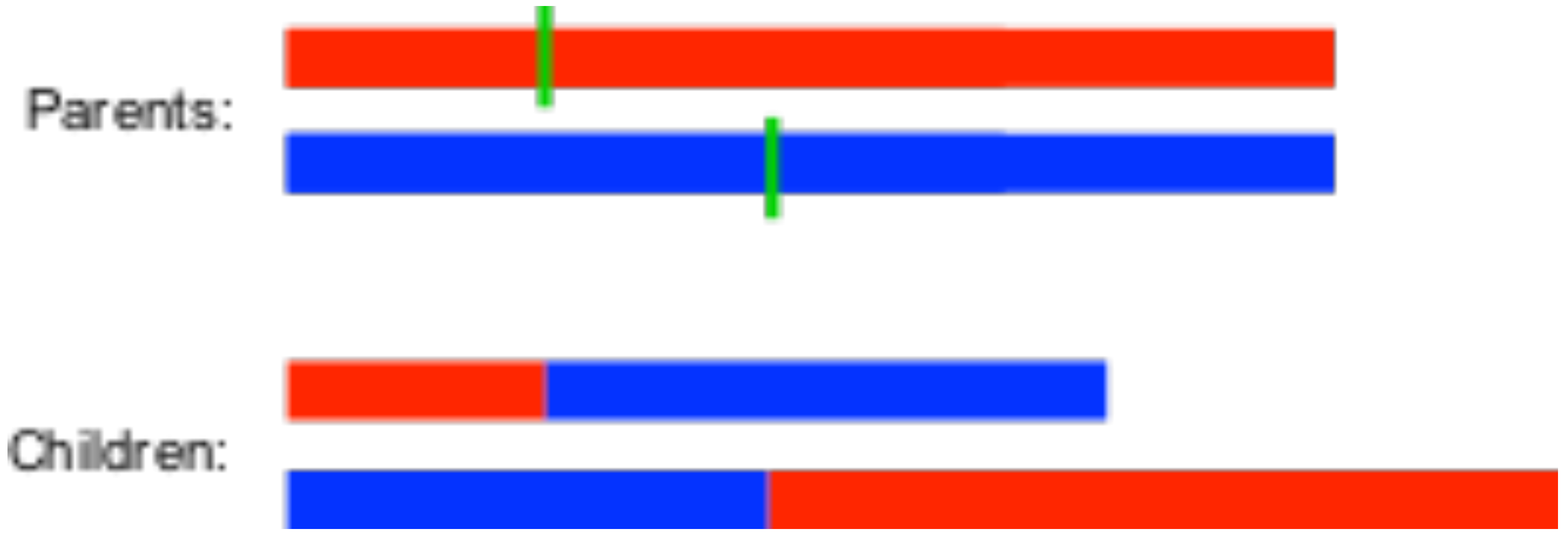
Selection

- Pick x individuals at random and find the best two

Mutation

- Replace a random point in the individual’s program with another operator or constant from the function set

Crossover



Questions and Programming Time!

Evolution Pseudo Code

```
Generate a population and set their population
While success hasn't been met
    Evaluate fitness on everybody
    While new generation is smaller than last generation
        Pick parents
        Make a new kid
        Put it in the population
    End
End
```