Genetic Algorithms

Alok Singh and Michael Kelly

Please record

What

Using ideas from *evolution* and *mutation* to solve problems.

Why

- Really general
- It's parallelizable
- Simple to implement
- It works.

How

Create

`individual`: a potential solution for the problem

`population`: randomly created set of individuals

Score

`fitness`: a function that says how close an individual is to being a solution

`score`: average loss

Cull

`cull`: survival of the fittest. Kill off all but some percentage of the fittest individuals and some lucky survivors

Evolve

`breed`: use `N` individuals to create a new individual

`mutate`: sometimes, individuals get mutations

`create_new_generation`: breed until population is back up to size

`evolve`: repeat until fitness is high enough for your liking

This framework is generic. Only the fitness function and the ways of creating and breeding individuals are problem-specific.

Michael and I used it to sum lists and to optimize neural network sizes.

Issues

- depends a lot being clever about breeding and mutation
- gets stuck in local optima

Feedback

`alok.blog/about`