Evolutionary Robotic Philosophy and Design Principles

By: Travis Hite

Keywords

Individuals: A specific instance of a training variable
Population: A group of individuals (mu)
Offspring: New individuals created from individuals in the population (lambda)
Fitness: How well an individual is doing
Hits: Negative feedback
Weight: Generally a number between 0 and 1 used to find a desired output strength
Binary Encoded: 01110010
Real Encoded: {-30,-30,-30,1}-{30,30,30,512}

A Design Perspective

Divide and conquer - Perception, planning, action
Building blocks - build layers upon layers
Distal vs Proximal descriptions of behavior
Genotype vs Phenotype descriptions of response
Example Scenario: Explore, avoid walls, approach target, discriminate target from wall

Evolution Problem

Natural Evolution and reproduction
Survival of the fittest leads to Bootstrap Problem
Solutions:

- More experimenter insight
- Incremental Evolution - Simple to Complex
- Self-Organized Incremental Evolution

The Basics of Genetics

Genetic Algorithms - Artificial chromosomes randomly modified repeated over generations.

Fitness - The higher the better

Selective Reproduction - Roulette wheel

- One point, two-point, multipoint Crossover (sexual)
- One point, multi-point, sign Mutation (asexual)

Schema Theory

Schema - Template for a family of strings

1*1 = 101 and 111

N^3 schemata processed ("Implicit Parallelism")

Significant components written farther apart leads to a higher probability of being broken down

Artificial Evolution in Autonomous Systems

Goal is complex abilities through interaction with environment
Expected to survive on their own
Loose fitness functions for better adaptability

Neural Network

Layers: Input, Hidden/Internal, Output

Feedforward - Signals travel from input to output
Recurrent – Signals may travel within network

Signals travel independently on weighted channels
- Step - output is either 0 or 1 dependent on threshold
- Linear - graded input with slope k
- Sigmoid - squashed between 0 and 1 with slope k

Learning Rates

Supervised Learning - synaptic strengths modified by difference between desired output and output given

Unsupervised Learning - Updates weights based on input value only.

A new learning rate is derived by taking the old weight and adding a new modification weight to it times a small learning rate between 0 and 1

Justifications For Evolving Neural Networks

- Smoother search space
- Varying evolutionary granularity
- Straightforward mapping from sensors to motor
- Robust to noise
- Biologically plausible
- GAs explore populations of networks, not singular.
- No constraints on type of architecture
- Detailed specifications of network not needed

Lisp and Genetic Programming

Genetic Programming - Encode the solution not the problem

Based on Lisp expressions

(+,2(*,3,2))=2+(3*2)

F={+,-,*,%,IFLTE}

T={X,Y,Z,R}

Above functions and terminals spliced together and mutated over generations.

Questions? Comments?

Hope I was interesting.