Computational Intelligence

Instructor
Tom Poliquin

Brief Introduction

Name

Where you Work

What you Expect from Class

What is your Favorite Language

Logistics

Class Structure

- Lecture
- In Class Problems
- Homework

Grading

Attendance 25%

Homework 50%

Class Participation 25%

What this course is

High Level Overview of CI

Emphasis on Practical Use

What this course is not

Theory

Systems Design

Statistical Problem Solving

Course Overview

- Definition
- History
- Problem Solving
- Neural Nets
- Genetic Algorithms
- Genetic Programming
- PSO
- Statistics

What is CI

Make Computers Think

Makes Possible Perception

Reasoning and Decision Making

Design of Intelligent Agents

History of CI

1956 Begins

(Central Planning)

1987 – Becomes Science

(More Agent Based)

History of CI

Recent Successes

- Medical Diagnosis
- Darpa Logistics
- Medical Robot Assistants
- Deep Blue

History of CI

Recent Successes

CI is a Piece
of the Solution

The Spice of the Meal

Problem Solving

Why do we need CI

Problems are Hard

Problem Solving

Hard Problems

Mathematically Difficult
Huge Search Spaces
Noisy
Ill Defined

*(Simplification often destroys Problem)

Problem Solving

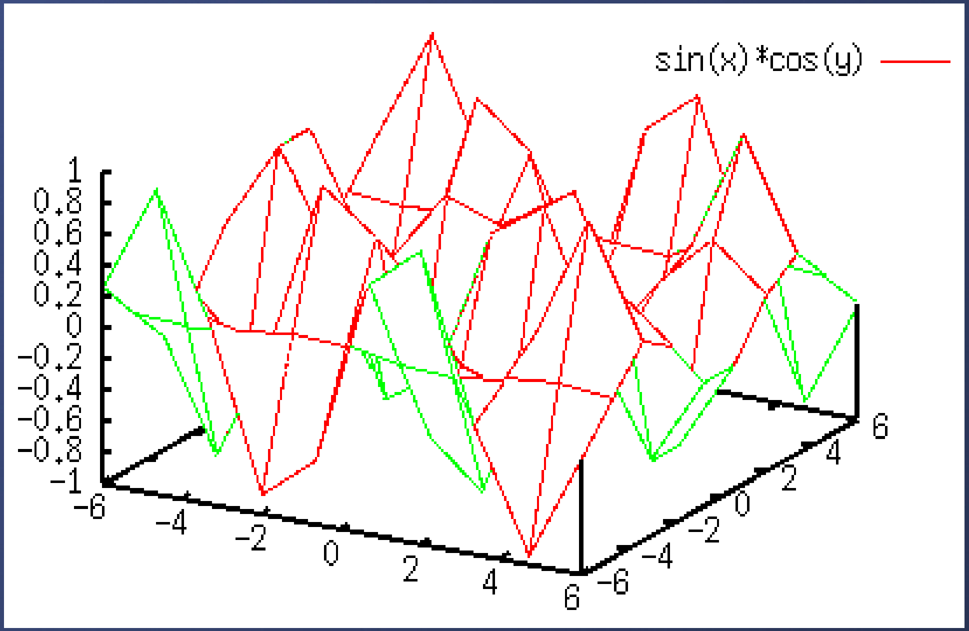
Dimensionality

How many Variables

Landscape

Typically Fitness is Z Direction

Hill Climbing



Problem Solving

Why are Problems Hard

- Large Search Space

- So Complicated that to get any Answer Model must be Oversimplified

- Evaluation Function Difficult

- The Person Solving Problem is unprepared or Imagines a Barrier to Solution

TSP Problem

Traveling Salesman

- List of Cities to Visit $\frac{(n-1)!}{2}$

- Find Shortest Path to Visit Each City only Once

10 Cities 181,000 Possible Solutions
20 Cities 10,000,000,000,000,000

SAT Problem

Boolean Satisfy Ability

What x's Makes F(x) True

$F(x) = (x_3 \text{ and } x_4) \text{ or } (x_1 \text{ and } x_2)$

What Happens with 100 Variables
(100 Dimensions)

1,000,000,000,000,000,000,000,000,000,000
Possibilities

Problem Solving

Traditional

Exhaustive Search   Dumb.

- - - |

- - -   V

Simulated Annealing   Smart

EC

Has Competition Among Solutions

Practical Concerns

Expert systems 80's

- Hard to Gather Data

- Static

- Politics

Practical Concerns

Neural Networks 90's

- Fragile Models
- Difficult to Maintain

Practical Concerns

CI Euphoria

By Applying Combinations of Techniques More Problems Can Be Solved

Practical Concerns

Dow Chemical Case Study

- Evolutionary Computation
- Symbolic Regression (GP/PSO)
  Allows Insight
- Optimization (GA/PSO)
- Neural Networks
- Clustering (K Means)
- Statistics

Practical Concerns

Dow Modeling Process

- Problem and Success Definition
- Data Preprocessing & Classification
- Variable Selection
- Data Condensation
- Model Generation (Genetic Programming)
- Model Selection
- Model Validation (Statistics)
- Model Exploitation
- Model Maintenance & Support

EC Basics

Design the Best Rabbit

Rabbit/Fox Paradigm	EC Paradigm
Rabbits	Population of Possible Solutions
Rabbit Survives Against Foxes	Fitness Function
Cosmic Rays	Variation
Surviving Rabbits Procreate	Selection

Designing Evolutionary Algorithms

- Representation
- Evaluation
- Variation
- Selection
- Initialization

EC Algorithm

Procedure Evolutionary Algorithm

```
begin
    t <- 0
    Initialize P(t)
    Evaluate   P(t)
    while (not terminating condition) do
    begin
        t <- t+1
        select P(t) from P(t-1)
        alter P(t)
        evaluate P(t)
    end
end
```

EC Practical Steps

Model	What Data do I have What can I Ignore
Objective	What Answer do I want
Representation	For my selected EC Paradigm, how should I Represent the problem
Fitness	How do I Decide which of the Solutions is better

EC - Details

Wouldn't it be nice
if we had a
“General Purpose Problem Solver?”

We Can't.

But we can use our
Domain Knowledge

NFL

No Free Lunch

No Algorithm Better
Than Another Averaged
Over all Problems

NFL

Our Algorithm

Exit a Room in the Dark

- Move in Straight Line Until Wall

- Move Along Wall Until Opening Felt

- Go Through Opening

NFL

Mr. NFL's Algorithm

Stop in all Corners

- Move in Straight Line Until Wall

- Move Along Wall Until Feel Corner

- Stop

NFL

Mr. NFL

"Averaged over all
Problems my Algorithm
is as good as yours"

- Find the Center of the Room

- Avoid all Walls

NFL

Some Algorithms are better than others for specific real world Problems

Must use knowledge about the Problem (Domain Specific)

EC Details

Representation

Possible Solutions -> Data Structure

Common Representations

- Fixed Length Vector
- Permutations
- FSM
- Symbolic Expressions

EC Details

Representation

Possible Solutions -> Data Structure

Common Representations

- Fixed Length Vector
- Permutations
- FSM
- Symbolic Expressions

EC Details

Representation

Fixed Length Vector

- Which to choose — List of Binary Strings
- Parameter Optimization — List of Floating Point Numbers
- Time Sequence — $[a_1, a_2, ...k]$

$y(t) = a, y[t-1] ... t$ any $[t-k]$

EC Details

Representation

Permutations [1, 2, 3, ... j]
Optimize Sequence Order

FSM (Finite State Machine)

[Number of States
For Each State
Input
Output
Next State
Starting State]

EC Details

Representation

Symbolic Expressions

Parse Tree

Functions
(Non-Leaf Nodes)
(+, -, * ...)
Terminals
(Leaf Nodes)
(Inputs, 2, 3 ...)

EC Details

Evaluation

- Judges “Goodness” of Possible Solution
- Can sometimes be Relative among Individuals
- Generally the most time consuming
- Once you find out It's horrible stop Evaluating
- Quality often depends on Representation

EC Details

Variation

For Fixed Length Vectors

Binary	- Flip a Bit - Crossover
Floating Point Integers	- Crossover - Add Zero Mean Gaussian Distributed Numbers To Each Value

EC Details

Variation

For FSMs

- Add a State
- Delete a State
- Change a Start State
- Change an Output Symbol
- Change next State

EC Details

Evaluation

For Symbolic Expressions

- Crossover
- Mutate One Node
- Swap Arguments

EC Details

Selection

Stochastic	Roulette Wheel Tournament
Deterministic	- Choose n best - Choose Only n best from Offspring

GP

Terminals

Variables	Input Sensors State Variables
Constants	E.G. 3 or Nil
Functions with no Args	To Generate Side Effects (Change State)
Connectives	$Progn_2$, $Progn_3$.....

GP

Functions

Arithmetic Functions	*, +, - ...
Math Functions	sin(x) ...
Boolean Operators	and ...
Conditionals	if ...
Domain Specific	?
List for Returning Multiple Values	(Lists...)

GP

So the Representation Problem
is Picking Functions and Terminals

We still need to know how to
do the Evaluation (Fitness)

GP

Tableau

Similar to EC

Model
Objective
Terminals
Functions
Fitness

GP

Boolean II Multiplexer

View it as Binary Symbolic Regression Problem

GP

Boolean II Tableau

Objective	Find S Expression with some Output as Function
Terminals	A_0, A_1, A_2 D_0, D_1, D_2, D_3, D_4, D_5, D_6, D_7
Functions	and, or, not, if
Fitness	# of cases (of the 2048) S-Expression is correct

GP

Bang – Bang Cart

Control System Problem

Bang-Bang Cart Tableau

Objective	Find Time Optimal Control Strategy to Center a Cart
Terminals	State Variables X and V
Functions	+, - , *, %, ABS, GT
Fitness	Time for 20 Fitness Cases to Center Cart Timeout Costs 10 Seconds

GP

Inverse Kinematics

$$x = f(\Theta_0, \Theta_1)$$

$$y = f(\Theta_0, \Theta_1)$$

Given x&y Find Θ_0, Θ_1

Inverse Kinematics Tableau

Objective	Find a Vector of 2 Angles to move a 2 Link arm to a Given x,y Position
Terminals	T_0 – x, y, ERC T_1 – Angle-0, x, y, ERC
Functions	List2, +, -, *, % Exp Asin, Acos, Atan (Root is Always List2)
Fitness	Sum over 25 Fitness Cases of the Error Distance

GP

Try	Broom Balancing
Find	Control Strategy to Balance Broom and Bring Car to Rest in Minimal Time

PSO

Particle Swarm Optimization

Social Behavior as Optimization

Boids

Ants

Bees

CAS Example

PSO

Concept for Optimizing Non-Linear Functions

Roots in Artificial Life and Evolutionary Computation

Effective on a wide Variety of Problems

PSO

Discovered Through Social Model Simulation

Related to Bird Flocking and Swarming Theory

Expanded to Multidimensional Search

Paradigm Simplified

PSO

$$V_{ID} = V_{ID} + C_1 \, rand \, (P_{ID} - X_{ID}) + C_2 \, rand \, (P_{GD} - X_{ID})$$

$$X_{ID} = X_{ID} + V_{ID}$$

$$C_1 = C_2$$

V is Limited to VMax

PSO

Neighborhoods

- All Particles

- Various Topologies
  Star, Ring, ETC

PSO

Similarities to EC

- Population of Solutions
- Randomly Initialized
- Interactions Among Population members

PSO

Differences from EC

- Solutions Flown Through Problem Space
- Each Individual Remembers its Best
- Individuals Survive

PSO

Changes to Original Algorithm

- Inertia
- VMax
- Changing Topologies
- Binary Version

PSO

Most Promising Application
is Training Neural Networks