



Computer Vision

Genetic Algorithms

27 August 2008

Copyright © 2001 – 2008 by
Noordelijke Hogeschool Leeuwarden and Van de Loosdrecht Machine Vision
All rights reserved

j.van.de.loosdrecht@tech.nhl.nl, jaap@vdlmv.nl

Genetic Algorithms (GA)

Overview:

- **Introduction**
- **Example: Mother Nature**
- **Example: Finding best fit for a polynomial**
- **Example: Camera calibration**
- **Overview Operators**
- **Example Polynomial using fitness all function (*)**
- **Example: Travelling salesman**
- **Example: KnapSack**
- **Example: Optimising learn error Neural Network**
- **Example: Optimising Neural Network with evaluation set**
- **Example: Optimising Neural Network selecting features**

Introduction

Idea:

In a population of beings the “strongest” are able to mate and to produce offspring. Disabled or sick beings have less opportunities to mate and have a smaller chance to produce healthy offspring.

When environmental conditions are changing only offspring which can adapt to the new situation have good changes to survive and produce adapted offspring.

In short: “survival of the fittest”

28-aug-08

Genetic Algorithms

3

Example Mother Nature

DNA represent the blueprint of a being

DNA mother: 11011010111010101001

DNA father: 10011010110001001011

Mating, DNA child:

- Crossover: 11011010110001001011
- Mutation: 100111101101001011

Mutation makes it possible to get children that are better then their parents and/or can adapt to a changing environment

Fitness function determines whether a being can reproduce

28-aug-08

Genetic Algorithms

4

Genetic Algorithms (GA) for optimisation problems

GA can be used for optimisation problems if for the solution domain can be defined:

- a genetic representation
- a fitness function

28-aug-08

Genetic Algorithms

5

Genetic Algorithm (GA) for optimisation problems

Pseudo code:

- Choose initial (random) population
- Evaluate the fitness of each individual in the population
- While not good enough best individual do
 - Select best ranking individuals to reproduce
 - Breed new generation through crossover and mutation
 - Add new generation to population
 - Evaluate the fitness of each individual in the population

28-aug-08

Genetic Algorithms

6

Example: finding best fit for a polynomial

- Generate a point cloud of measurements (x,y)
- Use GA to find parameters a, b, and c of polynomial

$$y = ax^2 + bx + c$$

- Solution domain:
 - Genetic representation:
the floating point numbers a, b and c
 - Fitness function:
the sum of errors of applying the solution to the point cloud

28-aug-08

Genetic Algorithms

7

Demonstration: finding best fit for a polynomial

- Open scripts:
 - ga_func.jls: calculation of the polynomial
 - ga_example.jls: the main program
 - ga_fitness.jls: calculation the fitness of a solution
 - Open variable window
 - Run script ga_example.jls
-
- Change line to create ga in:
`ga_create ga ga_fitness 10 &$paramTab 10 tracefile.txt`
 - Run script for about 20 generations
 - Examine file tracefile.txt to see the best 10 solutions for each generation
Note: if tracefile indicates that n best solution are (for a large part) the same the best solution is found or the Genetic Algorithm is stuck in a local minimum.
If the algorithm is stuck one or more of the parameters have to be changed

28-aug-08

Genetic Algorithms

8

ga_func.jls

```
// y = a * x**2 + b * x + c
return %p1 * %p4 * %p4 + %p2 * %p4 + %p3
%p1 = a
%p2 = b
%p3 = c
%p4 = x
function result = y
```

The screenshot shows the VisionLab interface with the script file open. The code defines a function `y` that calculates a quadratic equation based on parameters `a`, `b`, and `c`. The parameters are assigned values from `%p1` to `%p4`. A final line `function result = y` is present. The interface includes tabs for `ga_example.jls` and `ga_fitness.jls`, and various execution controls.

ga_example.jls

```
lAddScript ga_func ga_func.jls
lAddScript ga_fitness ga_fitness.jls
$trainSet = 10
InitRandomGen 1
for $i = 0 to $trainSet do
    $xTab[$i] = Random -2 2
    $yTab[$i] = icall ga_func 2 3 4 $xTab[$i]
endfor

//           low      high     delta     micro   mutationP deltaP
$paramTab[0] = -10      10      0.1    0.001    0.1      0.5
$paramTab[1] = -10      10      0.1    0.001    0.1      0.5
$paramTab[2] = -10      10      0.1    0.001    0.1      0.5

ga_create ga ga_fitness 10 &$paramTab 0 tracefile.txt
for $gen = 1 to 50 do
    //           maxGen minError deltaError microP
    $_error = ga_optimize ga 1      0.1      1      0.5
    $_sol = ga_getsolution ga
    SyncVars
endfor
ga_delete ga
```

The screenshot shows the VisionLab interface with the `ga_example.jls` script. The code initializes a parameter table with values for `low`, `high`, `delta`, and `micro` parameters. It then creates a genetic algorithm object and performs optimization. Red annotations highlight parts of the code: "Generate point cloud" and "Initialize parameter description table". The interface includes tabs for `ga_func.jls` and `ga_fitness.jls`, and various execution controls.

ga_example.jls

```

VisionLab V3.37 11-12-2007 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2007c\images\ga_example.jls
Internal script name Search Find Line number 19 Jump Indentation
lAddScript ga_func ga_func.jls
lAddScript ga_fitness ga_fitness.jls
$trainSet = 10
InitRandomGen 1
for $i = 0 to $trainSet do
    $xTab[$i] = Random -2 2
    $yTab[$i] = icall ga_func 2 3 4 $xTab[$i]
endfor

//      low      high      delta      micro      mutationP deltaP
$paramTab[0] = -10      10      0.1      0.001      0.1      0.5
$paramTab[1] = -10      10      0.1      0.001      0.1      0.5
$paramTab[2] = -10      10      0.1      0.001      0.1      0.5

ga_create ga ga_fitness 10 &$paramTab 0 tracefile.txt
for $gen = 1 to 50 do
    //      maxGen minError deltaError microP
    $_error = ga_optimize ga 1      0.1      1      0.5
    $sol = ga_getsolution ga
    SyncVars
endfor
ga_delete ga

```

Execute F9 | Abort Ctrl+Y | Protected | Continue | Single step | Step into | Reset

28-aug-08 11

ga_example.jls

```

VisionLab V3.37 11-12-2007 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2007c\images\ga_example.jls
Internal script name Search Find Line number 19 Jump Indentation
lAddScript ga_func ga_func.jls
lAddScript ga_fitness ga_fitness.jls
$trainSet = 10
InitRandomGen 1
for $i = 0 to $trainSet do
    $xTab[$i] = Random -2 2
    $yTab[$i] = icall ga_func 2 3 4 $xTab[$i]
endfor

//      low      high      delta      micro      mutationP deltaP
$paramTab[0] = -10      10      0.1      0.001      0.1      0.5
$paramTab[1] = -10      10      0.1      0.001      0.1      0.5
$paramTab[2] = -10      10      0.1      0.001      0.1      0.5

ga_create ga ga_fitness 10 &$paramTab 0 tracefile.txt
for $gen = 1 to 50 do
    //      maxGen minError deltaError microP
    $_error = ga_optimize ga 1      0.1      1      0.5
    $sol = ga_getsolution ga
    SyncVars
endfor
ga_delete ga

```

Execute F9 | Abort Ctrl+Y | Protected | Continue | Single step | Step into | Reset

28-aug-08 12

ga_fitness.jls

```

VisionLab V3.37 11-12-2007 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2007c\images\ga_example.jls
Internal script Search Find Line number Indentation
name: Store Jump + . c
C:\VisionCursus\Vision2007c\images\ga_fitness.jls
Internal script Search Find Line number Indentation
name: Store Jump + . c
$total = 0
$max = GetSizeArray &&Tab
$max = $max - 1
for $i = 0 to $max do
    $y_ga = icall ga_func %p2 %p3 %p4 &&Tab[$i]
    $e = $yTab[$i] - $y_ga
    $e = Fab $e
    $total = $total + $e
    if $total > %p1 then
        return $total
    endif
endfor
return $total

```

Calculate the total error
 of the point cloud
 %p1 = cutoff error
 %p2 = a
 %p3 = b
 %p4 = c

Execute F9 | Abort Ctrl+Y | Protected | Continue | Single step | Step into | Reset

0 us 1.67811

28-aug-08 Genetic Algorithms 13

ga_fitness.jls

```

VisionLab V3.37 26-1-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2008c\new_GA\ga_example.jls
Internal script Search Find Line number Indentation
name: Store Jump + . c
C:\VisionCursus\Vision2008c\new_GA\ga_fitness.jls
Internal script Search Find Line number Indentation
name: Store Jump + . c
// script: ga_fitness.jls
// optimising polynomial with GA
// Jaap van de Loosdrecht, 26-2-2008

$total = 0
$max = GetSizeArray &&Tab
$max = $max - 1
for $i = 0 to $max do
    $y_ga = icall ga_func %p2[0] %p2[1] %p2[2] &&Tab[$i]
    $e = $yTab[$i] - $y_ga
    $e = Fab $e
    $total = $total + $e
    if $total > %p1 then
        return $total
    endif
endfor
return $total

```

Calculate the total error
 of the point cloud
 %p1 = cutoff error
 %p2[0] = a
 %p2[1] = b
 %p2[2] = c

Execute F9 | Abort Ctrl+Y | Protected | Continue | Single step | Step into | Reset

0 us

28-aug-08 Genetic Algorithms 14

Run script ga_example

```

VisionLab V3.37 11-12-2007 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2007c\images\ga_example.jls Internal script Search
Variables a b c
name [ ] Store [ ]
_error = 0.07987
e = 0.00014
gen = 44
i=10
max = 10
paramTab = array(3) <-10 10 0.1 0.001 0.1 0.5 -10 10 0.1 0.001 0.1 0.5>
trainSet = 10
xTab = array(11) <-1.99499...-1.30357>
yTab = array(11) <5.975...3.48788>
y_ga = 3.48802
ga_create ga ga_fitness 10 &paramTab
for $gen = 1 to 50 do
// maxGen
    $error = ga_optimize ga 1
    $sol = ga_getsolution ga
    SyncVars
endfor
ga_delete ga

```

Close

stay on top Update Remove RemoveAll

Execute F9 Abort Ctrl+Y Protected Continue Single step Step into Reset

0 us 0.07987

28-aug-08 Genetic Algorithms 15

tracefile.txt

error	nr params	param 1	param 2	param 3
1.64561	3	2.021247383037812	2.951758323923461	4.123908902249214
1.64599	3	2.021247383037812	2.952424298837246	4.123908902249214
1.64624	3	2.021247383037812	2.952424298837246	4.123908902249214
1.64646	3	2.021247383037812	2.952424298837246	4.123908902249214
1.64634	3	2.021247383037812	2.952424298837246	4.123908902249214
1.6498	3	2.021683004242072	2.952147221289713	4.123698934904019
1.65048	3	2.021683004242072	2.952147221289713	4.123698934904019
0 iteration: 18				
1.63374	3	2.020741233558153	2.95130002746666	4.123524521622364
1.63453	3	2.020612018189031	2.952491500595112	4.123698934904019
1.64021	3	2.020612018189031	2.952491500595112	4.123698934904019
1.64046	3	2.020612018189031	2.952491500595112	4.123698934904019
1.64119	3	2.021301278725546	2.951512741477707	4.123441267128514
1.64131	3	2.021301278725546	2.951512741477707	4.123441267128514
1.64139	3	2.021301278725546	2.951512741477707	4.123441267128514
1.64335	3	2.021459242530594	2.95130002746666	4.123441267128514
1.64387	3	2.021301278725546	2.951512741477707	4.123698934904019
1.64387	3	2.021301278725546	2.951512741477707	4.123698934904019
0 iteration: 19				
1.63169	3	2.020612018189031	2.952491500595112	4.123441267128514
1.6326	3	2.020612018189031	2.952491500595112	4.123524521622364
1.633	3	2.020612018189031	2.952491500595112	4.123524521622364
1.63314	3	2.020741233558153	2.95130002746666	4.123441267128514
1.63374	3	2.020741233558153	2.95130002746666	4.123524521622364
1.63374	3	2.020741233558153	2.95130002746666	4.123524521622364
1.63406	3	2.020612018189031	2.951512741477707	4.123698934904019
1.63406	3	2.020612018189031	2.951512741477707	4.123698934904019
0 iteration: 20				
1.61835	3	2.019693868831446	2.952491500595112	4.123441267128514
1.62548	3	2.019955494247261	2.95130002746666	4.123839259010589
1.62702	3	2.020612018189031	2.952491500595112	4.123524521622364
1.62718	3	2.020612018189031	2.952491500595112	4.123524521622364
1.62775	3	2.02065752128666	2.951014831995607	4.123099124118778
1.62806	3	2.020741233558153	2.951762993255411	4.122969237342448
1.63108	3	2.020612018189031	2.951512741477707	4.123441267128514
1.63106	3	2.020612018189031	2.95130002746666	4.123441267128514
1.63118	3	2.020612018189031	2.951512741477707	4.123441267128514

28-aug-08 Genetic Algorithms 16

Example: Camera calibration

This example is from the chapter about camera calibration
See this chapter for more details

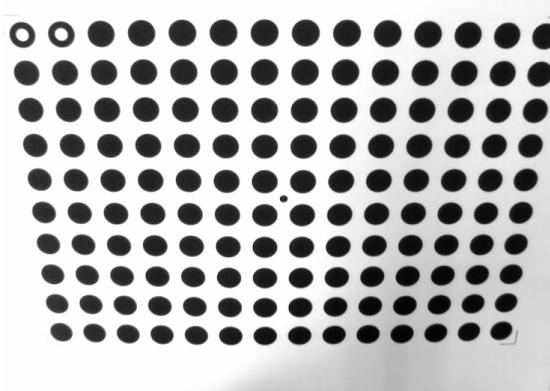
28-aug-08

Genetic Algorithms

17

Calibration Pattern

A fixed pattern of circles is used to calibrate the camera



28-aug-08

Genetic Algorithms

18

Camera Calibration

The 140 calibration points are used to find optimal values for the 15 camera parameters.

**So there are 140 equations with 15 unknown parameters.
The parameter space has 15 dimensions and consists of polynomial and goniometric equations.**

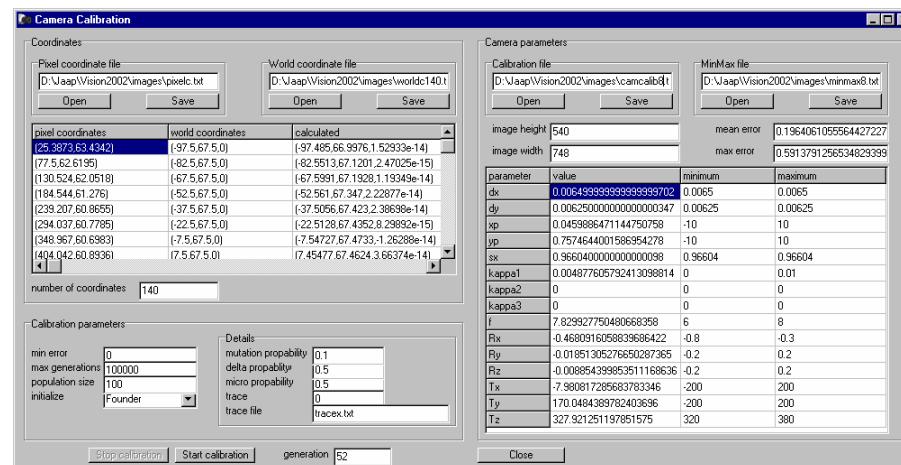
This non-linear optimisation problem is solved in VisionLab with a combination of a Genetic Algorithm and Hill Climbing.

28-aug-08

Genetic Algorithms

19

Demonstration Camera Calibration



28-aug-08

Genetic Algorithms

20

GA in VisionLab

VisionLab uses a combination of genetic algorithm and "inverted hill climbing" to descend to the bottom of the local minima
All parameters are represented as floating point values

Overview of operators:

- **GA_Create**
- **GA_Delete**
- **GA_RandomInitialize**
- **GA_SetFounder**
- **GA_SetFounders**
- **GA_Optimize**
- **GA_GetSolution**
- **GA_GetPopulation**

28-aug-08

Genetic Algorithms

21

GA_Create

**GA_Create (gaName, fitnessName, popSize,
trace, traceFileName, paramTab)**

Create an instance of a Genetic Algorithm

Parameters:

- **gaName:** name of variable
- **fitnessName:** name of internal script
 %p1 is cutoff error
 %p2 is the name of an array variable without the '\$' containing
 the parameters to optimise, %\$p2[0] is the first parameter,
 %\$p2[1] is the second parameter, etc
- **popSize:** the population size
- **trace:** = 0 then no tracing, > 0 = number of best in population
 to trace
- **traceFileName:** filename to which trace log is written

28-aug-08

Genetic Algorithms

22

GA_Create

Parameters continued:

- paramTab: array with the description of the parameters, one line for each parameter:
 - low: the lowest possible value for the parameter
 - high: the highest possible value for the parameter
 - delta: step size used in finding the bottom in local minimum
 - micro: step size used by selfmating
 - mutationP: probability for mutation when mating, range [0..1]
 - deltaP: determines how much time will be used be used for "inverted hill climbing", range [0..1]

28-aug-08

Genetic Algorithms

23

GA_CreateAll

**GA_CreateAll (gaName, fitnessName, popSize,
trace, traceFileName, paramTab, allTab)**

Create an instance of a Genetic Algorithm for with the fitness function will be called one time for each generation.

28-aug-08

Genetic Algorithms

24

GA_CreateAll

Parameters:

- **gaName:** name of variable
- **fitnessName:** name of internal script
 %p1 is cutoff error
 %p2 is the name of an array variable without the '\$' containing the parameters for each being to optimise.
 \$%p2[0] contains the parameters for the first being, \$%p2[1] for the second being etc.
 The fitness script will have to insert the error for each being in front of the parameters for the being.
- **popSize:** the population size
- **trace:** = 0 then no tracing, > 0 = number of best in population to trace
- **traceFileName:** filename to which trace log is written

28-aug-08

Genetic Algorithms

25

GA_CreateAll

Parameters continued:

- **paramTab:** array with the description of the parameters, one line for each parameter:
 - **low:** the lowest possible value for the parameter
 - **high:** the highest possible value for the parameter
 - **delta:** step size used in finding the bottom in local minimum
 - **micro:** step size used by selfmating
 - **mutationP:** probability for mutation when mating, range [0..1]
 - **deltaP:** determines how much time will be used for "inverted hill climbing", range [0..1]

28-aug-08

Genetic Algorithms

26

GA_Delete

GA_Delete (gaName)

Delete an instance of a Genetic Algorithm

Parameter:

- **gaName: name of variable**

28-aug-08

Genetic Algorithms

27

GA_RandomInitialize

GA_RandomInitialize (gaName, populationSize)

Random intialise the population

Parameters:

- **gaName: name of variable**
- **populationSize: the population size**

28-aug-08

Genetic Algorithms

28

GA_SetFounder

GA_SetFounder (gaName, populationSize, founder)

Initialise the population with offspring from a parent

Parameters:

- **gaName: name of variable**
- **populationSize: the population size**
- **founder: the parameters of the founder**

28-aug-08

Genetic Algorithms

29

GA_SetFounders

GA_SetFounders (gaName, parents)

Parameters:

- **gaName: name of variable**
- **parents: an array with the parents, one line for each parent**

28-aug-08

Genetic Algorithms

30

GA_Optimize

**GA_Optimize (gaName, maxGenerations, minError,
deltaError, microP)**

**Use the genetic algorithm to find a best solution.
The function result is the fitness error of the best solution.**

Parameters:

- **gaName: name of variable**
- **maxGenerations: maximum number of generations**
- **minError: if fitness error < minError the optimization is stopped**
- **deltaError: minimum error for start of inverted hillclimbing**
- **microP: probability that self mate is a micro mutation**

28-aug-08

Genetic Algorithms

31

GA_GetSolution

GA_GetSolution (gaName)

Function result is the best solution found

Parameter:

- **gaName: name of variable**

28-aug-08

Genetic Algorithms

32

GA_GetPopulation

GA_GetPopulation (gaName, population)

Retreive the current population

Parameter:

- **gaName: name of variable**
- **population: array to store the population, one line for each parent**

28-aug-08

Genetic Algorithms

33

Demonstration: finding best fit for a polynomial using fitness all function

- Open scripts:
 - **ga_func.jls**: calculation of the polynomial
 - **ga_all_example.jls**: the main program
 - **ga_fitness_all.jls**: calculation the fitness of a solution
- Open variable window
- Run script **ga_all_example.jls**
- Note: in this example the fitness function will evaluate all beings instead of one being in the previous example.
- The only differences between **ga_example.jls** and **ga_all_example.jls** are:
 - **IAddScript ga_fitness ga_fitness_all.jls**
 - **ga_createall ga ga_fitness 10 &\$paramTab 0 tracefile.txt &\$allTab**

28-aug-08

Genetic Algorithms

34

ga_all_example.jls

```

lAddScript ga_func ga_func.jls
lAddScript ga_fitness ga_fitness_all.jls
$strainSet = 10
InitRandomGen 1
for $i = 0 to $strainSet do
    $xTab[$i] = Random -2 2
    $yTab[$i] = icall ga_func 2 3 4 $xTab[$i]
endfor

//           low      high      delta      micro      mutationP deltaP
$paramTab[0] = -10      10      0.1      0.001      0.1      0.5
$paramTab[1] = -10      10      0.1      0.001      0.1      0.5
$paramTab[2] = -10      10      0.1      0.001      0.1      0.5

ga_createall ga ga_fitness 10 &$paramTab 0 tracefile.txt &$allTab
for $_gen = 1 to 50 do
    //           maxGen minError deltaError microP
    $_error = ga_optimize ga 1      0.1      1      0.5
    $_sol = ga_getsolution ga
    SyncVars

```

28-aug-08 Genetic Algorithms 35

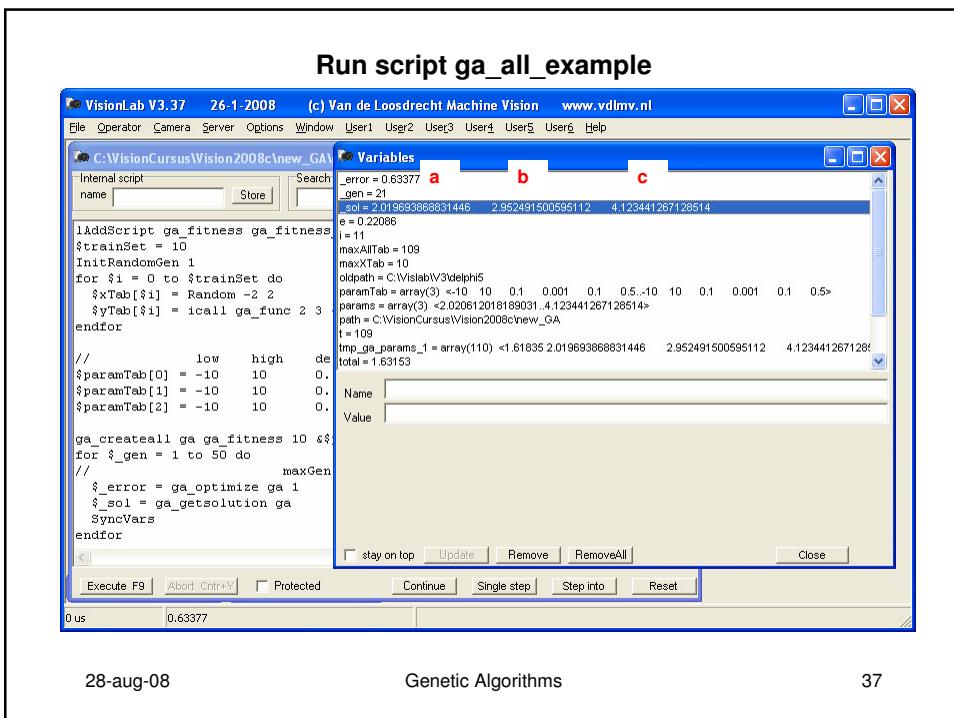
ga_fitness_all.jls

```

for $t = 0 to $maxAllTab do
    $total = 0
    VarToArrray &$%p2[$t] &$params
    for $i = 0 to $maxXTab do
        $y_ga = icall ga_func $params[0] $params[1] $params[2] $:
        $e = $yTab[$i] - $y_ga
        $e = Fabs $e
        $total = $total + $e
        if $total > %p1 then
            $i = $maxXTab + 1
            endif
        endfor
        $%p2[$t] = concat $total &$%p2[$t]
    endfor

```

28-aug-08 Genetic Algorithms 36



Example: Travelling Salesman Problem (TSP)

Optimizing problem:

- Given a number of cities and the distance between the cities, what is the least-cost round-trip route that visits each city and returns to the starting city?
 - Solution domain:
 - Genetic representation: a list indicating the order in which the cities are visited
 - Fitness function: total distance of the round-trip

Example: TSP

- Cities are random pixel chosen in an empty image
- Genetic representation:
 - a list indicating the order in which the cities are visited
 - VisionLab implementation works with floating point numbers
 - an array with floating point values indicating the order in which the cities are visited list, $0 \leq \text{value} \leq 1$, lowest values are visited first
- Fitness function:
 - Euclidian distance between cities

28-aug-08

Genetic Algorithms

39

Demonstration: TSP

- Open scripts:
 - ga_travel.jls: the main program
 - ga_travel_costfunc.jls: calculation of the fitness of a solution
 - ga_travel_sort.jls: generate a route
- Open variable window
- Run script ga_travel.jls

28-aug-08

Genetic Algorithms

40

ga_travel.jls (3)

```

VisionLab V3.37 8-3-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2008c\new_GA\ga_travel.jls
Internal script name | Store | Search | Find | Line number 44 Jump | Indentation + - c |
laddscript ga_travel_costfunc ga_travel_costfunc.jls
laddscript ga_travel_route ga_travel_route.jls
RemoveVar $$paramtab
// <low> <high> <delta> <micro> <mutationP> <deltaP>
for $i = 0 to $count
    $$paramtab[$i] = 0 1 0 0.1 0.2 0
endfor

GA_Create ga ga_travel_costfunc 6 $$paramtab 1 tracefile.txt
while true
    $$dist = GA_Optimize ga 1 0 0 0.5
    $$sol = GA_GetSolution ga
    vartoarray $$sol $$sol_arr
    icall ga_travel_route $$sol_arr $$x_src $$y_src $$x $$y
    // display result
    copy img img_disp
    setallpixels img_disp 0
    $count = $cities - 1
    for $i = 1 to $count
        $ii = $i - 1
        $$c_src = vector $$x[$ii] $$y[$ii]
        $$c_dst = vector $$x[$i] $$y[$i]
    endfor
    drawline img_disp $$c_src $$c_dst 3 KeepOriginal
    $$c_src = vector $$x[$count] $$y[$count]
    $$c_dst = vector $$x[0] $$y[0]
    drawline img_disp $$c_src $$c_dst 3 KeepOriginal
    or img_disp imgcities
    textonimage img_disp 10 10 courier10 $$dist
    display img_disp
endwhile

```

0 us 1329.82

Initialize parameter description table

Generate n generations

28-aug-08 Genetic Algorithms 43

ga_travel.jls (4)

```

VisionLab V3.37 8-3-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2008c\new_GA\ga_travel.jls
Internal script name | Store | Search | Find | Line number 44 Jump | Indentation + - c |
$$dist = GA_Optimize ga 1 0 0 0.5
$$sol = GA_GetSolution ga
vartoarray $$sol $$sol_arr
icall ga_travel_route $$sol_arr $$x_src $$y_src $$x $$y
// display result
copy img img_disp
setallpixels img_disp 0
$count = $cities - 1
for $i = 1 to $count
    $ii = $i - 1
    $$c_src = vector $$x[$ii] $$y[$ii]
    $$c_dst = vector $$x[$i] $$y[$i]
    drawline img_disp $$c_src $$c_dst 3 KeepOriginal
    $$c_src = vector $$x[$count] $$y[$count]
    $$c_dst = vector $$x[0] $$y[0]
    drawline img_disp $$c_src $$c_dst 3 KeepOriginal
    or img_disp imgcities
    textonimage img_disp 10 10 courier10 $$dist
    display img_disp
endwhile

```

0 us 1329.82

Generate route from GA solution

Display route in image

28-aug-08 Genetic Algorithms 44

ga_travel_costfunc.jls

```

// script: ga_travel_costfunc.jls
// generate list with cities in travel order
// Klaas Dijkstra, 11-3-2008

copyvar &$p2 &$arr
icall ga_travel_route $arr $x_src $y_src $$x $$y
$count = GetSizeArray &$arr
$count = $count - 1
$totaldist = 0
for $i1 = 1 to $count
    $i1 = $i1 - 1
    $dist = (( $$x[$i1] - $$x[$i] ) * ($$x[$i1] - $$x[$i]) + (( $$y[$i1] - $$y[$i] ) * ($$y[$i1] - $$y[$i])
    $totaldist = $totaldist + $dist
    if $totaldist > $p1
        return $p1
    endif
endfor
$dist = (( $$x[$count] - $$x[0] ) * ($$x[$count] - $$x[0]) + (( $$y[$count] - $$y[0] ) * ($$y[$count] - $$y[0])
$dist = sqrt $$dist
$totaldist = $totaldist + $dist
return $totaldist

```

28-aug-08 Genetic Algorithms 45

ga_travel_route.jls (1)

```

//%p1 = [input]          Genetic result
//%p2 = [input]          X source
//%p3 = [input]          Y source
//%p4 = [output]         Ordered list of X values
//%p5 = [output]         Ordered list of Y values
copyvar &%p1 &$arr
copyvar &%p2 &$xx
copyvar &%p3 &$yy
$count = GetSizeArray &$arr
$count = $count - 1
//Bubblesort
$sv = 1
$si = 0
while ( $si < $count ) and ( $sv == 1 )
    $sv = 0
    $sj = 0
    while ( $sj < ( $count - $si ) )
        $sj1 = $sj + 1
        if ( $arr[$sj] > $arr[$sj1] )
            $temp1 = $arr[$sj]
            $temp2 = $xx[$sj]
            $temp3 = $yy[$sj]
            $arr[$sj] = $arr[$sj1]
            $xx[$sj] = $xx[$sj1]
            $yy[$sj] = $yy[$sj1]
            $arr[$sj1] = $temp1

```

28-aug-08 Genetic Algorithms 46

ga_travel_route.jls (2)

```

$$v = 1
$$i = 0
while ( $$i < $$count ) and ( $$v == 1 )
  $$v = 0
  $$j = 0
  while ( $$j < ( $$count - $$i ) )
    $$j1 = $$j + 1
    if ( $$arr[$$j] > $$arr[$$j1] )
      $$temp1 = $$arr[$$j]
      $$temp1 = $$x[$$j]
      $$temp1 = $$y[$$j]
      $$arr[$$j] = $$arr[$$j1]
      $$x[$$j] = $$x[$$j1]
      $$y[$$j] = $$y[$$j1]
      $$arr[$$j1] = $$temp1
      $$x[$$j1] = $$temp2
      $$y[$$j1] = $$temp3
      $$v = 1
    endif
    $$j = $$j + 1
  endwhile
  $$i = $$i + 1
endwhile
copyvar $$x &p4
copyvar $$y &p5

```

28-aug-08 Genetic Algorithms 47

Running TSP (1)

```

while true
  $$dist = GA_Optimize ga 1 0 0 0.5
  $$sol = GA_GetSolution ga
  vartarray $$sol $$sol_arr
  icall ga_travel_route $$sol_arr $$x_src $$y_src $$x $$
  // display result
  copyimg img_disp
  setallpixels img_disp 0
  $count = $cities - 1
  for $i = 1 to $count
    $ii = $i - 1
    $c_src = vector $$x[$ii] $$y[$ii]
    $c_dst = vector $$x[$i] $$y[$i]
    drawline img_disp $c_src $c_dst 3 KeepOriginal
  endfor
  $c_src = vector $$x[$count] $$y[$count]
  $c_dst = vector $$x[0] $$y[0]
  drawline img_disp $c_src $c_dst 3 KeepOriginal
  or img_disp imgcities
  textonimage img_disp 10 10 courier10 $$dist
  display img_disp
endwhile

```

28-aug-08 Genetic Algorithms 48

Running TSP (2)

VisionLab V3.37 8-3-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl

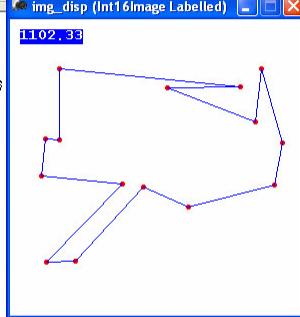
```

C:\VisionCursus\Vision2008c\new_GA\ga_travel.jls
Internal script name: Store Search Find Line number 44 Indentation
while true
    $&dist = GA_Optimize ga 1 0 0 0.5
    $sol = GA_GetSolution ga
    vartoarray $$sol $$sol_arr
    icall ga_travel_route $$sol_arr $x_src $y_src $$x $y
    // display result
    copy img img_disp
    setallpixels img_disp 0
    $count = $cities - 1
    for $i = 1 to $count
        $ii = $i - 1
        $c_src = vector $$x[$ii] $$y[$ii]
        $c_dst = vector $$x[$i] $$y[$i]
        drawline img_disp $c_src $c_dst 3 KeepOriginal
    endfor
    $c_src = vector $$x[$count] $$y[$count]
    $c_dst = vector $$x[0] $$y[0]
    drawline img_disp $c_src $c_dst 3 KeepOriginal
    or img_disp imgcities
    textonimage img_disp 10 10 courier10 $$dist
    display img_disp
endwhile

```

Execute F9 Abort Ctrl+Y Protected Continue Single step Step into Reset

0 us 1102.33



28-aug-08

Genetic Algorithms

49

Running TSP (3)

VisionLab V3.37 8-3-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl

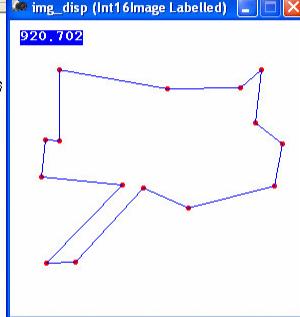
```

C:\VisionCursus\Vision2008c\new_GA\ga_travel.jls
Internal script name: Store Search Find Line number 44 Indentation
while true
    $&dist = GA_Optimize ga 1 0 0 0.5
    $sol = GA_GetSolution ga
    vartoarray $$sol $$sol_arr
    icall ga_travel_route $$sol_arr $x_src $y_src $$x $y
    // display result
    copy img img_disp
    setallpixels img_disp 0
    $count = $cities - 1
    for $i = 1 to $count
        $ii = $i - 1
        $c_src = vector $$x[$ii] $$y[$ii]
        $c_dst = vector $$x[$i] $$y[$i]
        drawline img_disp $c_src $c_dst 3 KeepOriginal
    endfor
    $c_src = vector $$x[$count] $$y[$count]
    $c_dst = vector $$x[0] $$y[0]
    drawline img_disp $c_src $c_dst 3 KeepOriginal
    or img_disp imgcities
    textonimage img_disp 10 10 courier10 $$dist
    display img_disp
endwhile

```

Execute F9 Abort Ctrl+Y Protected Continue Single step Step into Reset

0 us 920.702



28-aug-08

Genetic Algorithms

50

Example: KnapSack

Optimizing problem:

- Best choice of items that can fit in KnapSack
- KnapSack has maximum load in kg (= cost)
- Each item has a weight (kg) and a value
- “What is the maximum value V that can be achieved without exceeding the cost C”
- **Solution domain:**
 - Genetic representation:
a list of 0's or 1's indicating if item is in KnapSack
 - Fitness function:
total value of items in KnapSack not exceeding max weight

28-aug-08

Genetic Algorithms

51

Example: KnapSack

- **Genetic representation:**
a list of 0's or 1's indicating if item is in KnapSack
 - VisionLab implementation works with floating point numbers
 - **For each parameter (item):**
 - Low = 0
 - High = 2
 - item included = (Value \geq 1)

28-aug-08

Genetic Algorithms

52

Example: KnapSack

- **Fitness function:**
total value of items in KnapSack not exceeding max weight
 - **Fitness function:**
`if total weight <= max weight then
 total value of items in KnapSack
else
 0`
 - **Note:** this fitness function must be maximized
GA can only minimize a fitness function
 - **Redefined fitness function:**
`if total weight <= max weight then
 topValue - total value of items in KnapSack
else
 topValue`
 - Where **topValue > sum value of all items**

28-aug-08

Genetic Algorithms

53

Demonstration: KnapSack

- Open scripts:
 - knapsack.jls: the main program
 - knapsackCostFunc.jls: calculation of the fitness of a solution
- Open variable window
- Run script knapsack.jls

28-aug-08

Genetic Algorithms

54

knapsack.jls

```

VisionLab V3.37 11-12-2007 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2007c\Images\Knapsack.jls
Internal script name [ ] Store Search Find Line number [20] Jump Indentation [+ - .]
lAddScript costFunc knapsackCostFunc.jls
$maxWeight = 15
$stopValue = 1000
$nItems = 20
$maxIndex = $nItems - 1
InitRandomGen
for $i = 0 to $maxIndex do
    $wTab[$i] = Random 1 5
    $vTab[$i] = Random 1 5
endfor
//           low      high     delta     micro     mutationP   deltaP
$paramTab[0] = 0      2        1        1        0.2          0.5
for $i = 1 to $maxIndex do
    $paramTab[$i] = $paramTab[0]
endfor
ga_create ga costFunc 10 &$paramTab 0 tracefile.txt
for $gen = 1 to 50 do
    //           maxGen minError deltaError microP
    $error = ga_optimize ga 1      0        0        0.9
    $_maxValue = $stopValue - $error
    $sol = ga_getsolution ga
    VarToArray $sol &$sTab
    $_sack = ""
    $_weight = 0
    for $i = 0 to $maxIndex do
        if $_sTab[$i] > 1 then
            $s = 1
            $_weight = $_weight + $wTab[$i]
        else
            $s = 0
        endif
        $_sack = Concat $_sack $s
    endfor
    SyncVars
endfor
ga_delete ga

```

28-aug-08 Genetic Algorithms 55

knapsack.jls

```

VisionLab V3.37 11-12-2007 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
$wTab[$i] = Random 1 5
endfor
//           low      high     delta     micro     mutationP   deltaP
$paramTab[0] = 0      2        1        1        0.2          0.5
for $i = 1 to $maxIndex do
    $paramTab[$i] = $paramTab[0]
endfor
ga_create ga costFunc 10 &$paramTab 0 tracefile.txt
for $gen = 1 to 50 do
    //           maxGen minError deltaError microP
    $error = ga_optimize ga 1      0        0        0.9
    $_maxValue = $stopValue - $error
    $sol = ga_getsolution ga
    VarToArray $sol &$sTab
    $_sack = ""
    $_weight = 0
    for $i = 0 to $maxIndex do
        if $_sTab[$i] > 1 then
            $s = 1
            $_weight = $_weight + $wTab[$i]
        else
            $s = 0
        endif
        $_sack = Concat $_sack $s
    endfor
    SyncVars
endfor
ga_delete ga

```

28-aug-08 Genetic Algorithms 56

knapsackCostFunc.jls

```

// script: knapsackCostFunc.jls
// optimising knapsack with GA
// Jaap van de Loosdrecht, 26-2-2008
$paramTab[0] = 0
$cwd
$value = 0
$weight = 0
for $j = 0 to $maxIndex do
  if $%p2[$j] >= 1 then
    $weight = $weight + $wTab[$j]
  endif
  $value = $value + $vTab[$j]
endfor
return $topValue - $value

```

28-aug-08

Genetic Algorithms

57

Run knapsack.jls

```

// low high delta micro mutationP deltaP
$paramTab[0] = 0 2 1
for $i = 1 to $maxIndex do
  $paramTab[$i] = $paramTab[0]
endfor
ga_create ga costfunc 10 &$paramTab
for $gen = 1 to 50 do
  maxGen = 15
  maxIndex = 19
  maxWeight = 15
  nItems = 20
  paramTab = array(20) <0 2 1 1 0.2 0.5 0 2 1 1 0.2 0.5>
  s = 0
  $error = ga_optimize ga 1 0
  $maxValue = $topValue - $error
  $sol = ga_getsolution ga
  VarToArray &$sol &$tab
  $sack = ""
  $weight = 0
  for $i = 0 to $maxIndex do
    if $sol[$i] >= 1 then
      $s = 1
      $weight = $weight + $tab[$i]
    else
      $s = 0
    endif
    $sack = Concat $sack $s
  endfor
  SyncVars
endfor

```

28-aug-08

Genetic Algorithms

58

Example: Optimising learn error Neural Network

In this example the neural network for recognizing the OCR will be optimized from the chapter about Neural Networks

The training error will be used as fitness criteria

Parameters to optimize:

- Nr nodes hidden layer
- Learn rate
- Momentum

28-aug-08

Genetic Algorithms

59

Demonstration GA optimizing Neural Network

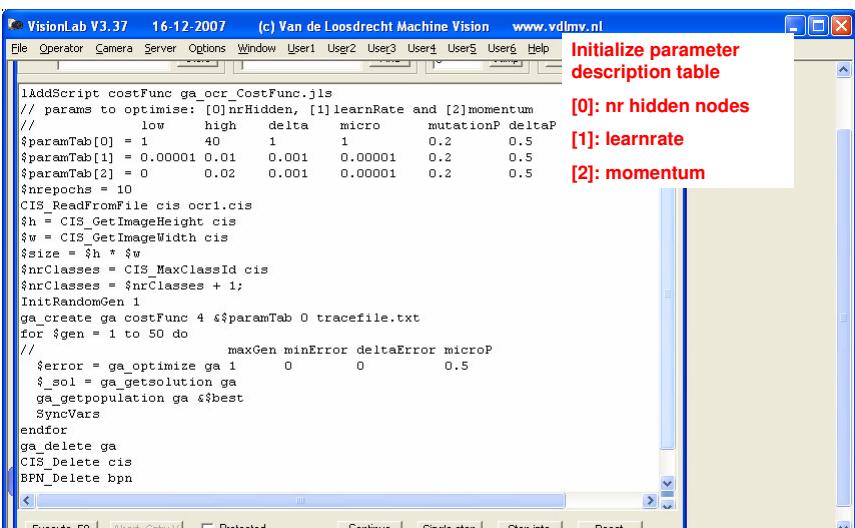
- Open scripts:
 - ga_ocr.jls: the main program
 - ga_ocr_CostFunc.jls: calculation the fitness of a solution
 - Open variable window
 - Run script ga_ocr.jls
-
- Note 1: this is time consuming, for real optimisation \$nrePOCHS and population size should be greater
 - Note 2: if calculations at server side takes much time the client will generate a timeout exception. The timeout value can be changed in the Server Options form in the Server menu.

28-aug-08

Genetic Algorithms

60

ga_ocr.jls



Initialize parameter
description table

[0]: nr hidden nodes

[1]: learnrate

[2]: momentum

```
File Operator Camera Server Options Window User1 User2 User3 User4 User5 Help
lAddScript costFunc ga_ocr_CostFunc.jls
// params to optimise: [0]nrHidden, [1]learnRate and [2]momentum
//           low      high    delta   micro   mutationP deltaP
$paramTab[0] = 1        40     1       1       0.2      0.5
$paramTab[1] = 0.00001  0.01   0.001   0.00001  0.2      0.5
$paramTab[2] = 0        0.02   0.001   0.00001  0.2      0.5
$nrEpochs = 10
CIS_ReadFromFile cis ocr1.cis
$h = CIS_GetImageHeight cis
$w = CIS_GetImageWidth cis
$size = $h * $w
$nrClasses = CIS_MaxClassId cis
$nrClasses = $nrClasses + 1;
InitRandomGen 1
ga_create ga costFunc 4 &$paramTab 0 tracefile.txt
for $gen = 1 to 50 do
//
//           maxGen minError deltaError microP
$error = ga_optimize ga 1      0      0      0.5
$_sol = ga_getsolution ga
ga_getpopulation ga &best
SyncVars
endfor
ga_delete ga
CIS_Delete cis
BPN_Delete bpn

```

Execute F9 | Abort Ctrl+Y | Protected | Continue | Single step | Step into | Reset |

0 us

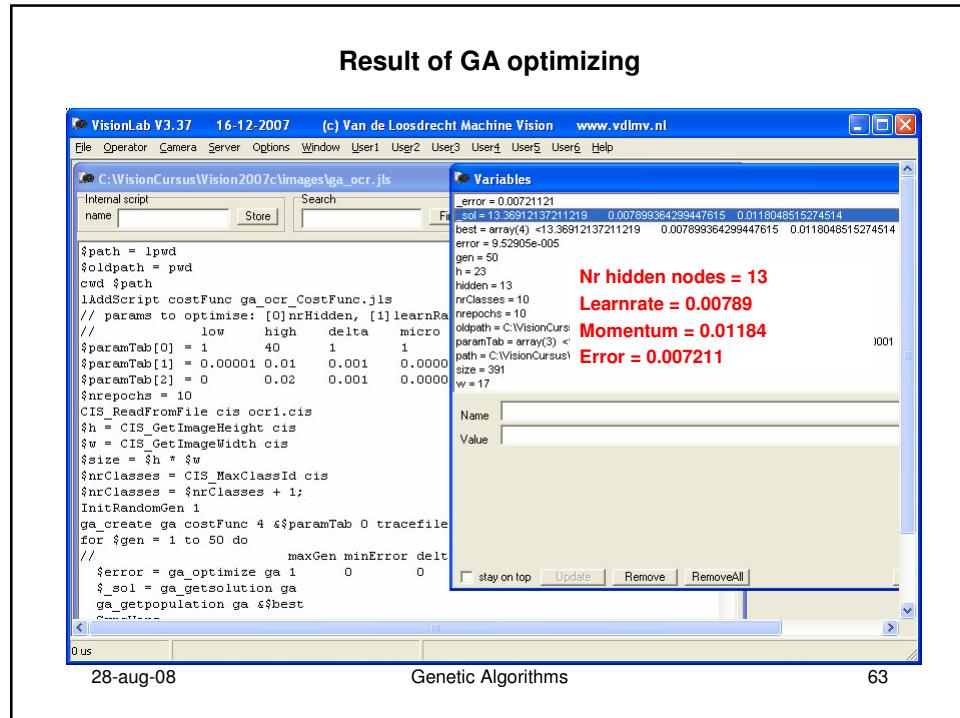
ga_ocr_CostFunc.jls

```
C:\VisionCursus\Vision2008c\new_GA\ga_ocr.jls
C:\VisionCursus\Vision2008c\new_GA\ga_ocr_CostFunc.jls

// script: ga_ocr_CostFunc.jls
// Optimising BPN with GA
// Jaap van de Loosdrecht, 26-2-2008

$hidden = FloatToInt $p2[0]
BPN_CreateImageClassifier bpn ByteImage $size $hidden 0 $nrClasses Bias 0 2!
$_error = BPN_TrainCIS bpn cis $nrePOCHS $p2[1] $p2[2]
RemoveFirstWord $_error
return $_error

Create and train Neural network
%p1 = cutoff error, not used
$p2[0] = nr hidden nodes
$p2[1] = learnrate
$p2[2] = momentum
Function result = training error
```



Example: Optimising Neural Network with evaluation set

In this example the neural network for recognizing the OCR will be optimized from the chapter about Neural Networks

The fitness criteria is chosen as:

- Number of false classifications * 1000 +
 - Number of low confidences +
 - Error in output layer

Parameters to optimize:

- Nr nodes hidden layer
 - Learn rate
 - Momentum

Demonstration Optimising Neural Network with evaluation set

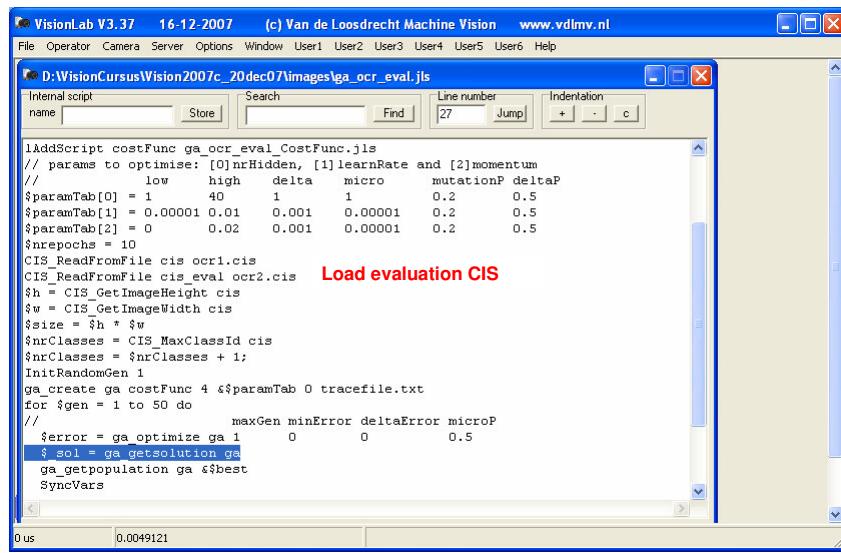
- Open scripts:
 - ga_ocr_eval.jls: the main program
 - ga_ocr_eval_CostFunc.jls: calculation the fitness of a solution
- Open variable window
- Run script ga_ocr_eval.jls
- Note 1: this is time consuming, for real optimisation \$nrePOCHS should be greater
- Note 2: if calculations at server side takes much time the client will generate a timeout exception. The timeout value can be changed in the Server Options form in the Server menu.

28-aug-08

Genetic Algorithms

65

ga_ocr_eval.jls



```

VisionLab V3.37 16-12-2007 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
D:\VisionCursus\Vision2007c_20dec07\images\ga_ocr_eval.jls
Internal script Search Find Line number 27 Jump Indentation + - c
name [ ] Store [ ] CIS ReadFromFile cis ocr1.cis
// params to optimise: [0]nrHidden, [1]learnRate and [2]momentum
//           low    high   delta   micro   mutationP deltaP
$paramTab[0] = 1     40     1       0.2     0.5
$paramTab[1] = 0.00001 0.01   0.001   0.00001 0.2     0.5
$paramTab[2] = 0     0.02   0.001   0.00001 0.2     0.5
$nrePOCHS = 10
CIS ReadFromFile cis_eval ocr2.cis Load evaluation CIS
$h = CIS_GetImageHeight cis
$w = CIS_GetImageWidth cis
$size = $h * $w
$nrClasses = CIS_MaxClassId cis
$nrClasses = $nrClasses + 1;
InitRandomGen 1
ga_create ga_costfunc 4 &$paramTab 0 tracefile.txt
for $gen = 1 to 50 do
  //          maxGen minError deltaError microP
  $error = ga_optimize ga 1 0 0 0.5
  $sol = ga_getsolution ga
  ga_getpopulation ga $best
  SyncVars

```

28-aug-08

Genetic Algorithms

66

ga_ocr_eval_CostFunc.jls

```

// script: ga_ocr_eval_CostFunc.jls
// optimising BPN with GA and eval cis
// Jaap van de Loosdrecht, 26-2-2008

$hidden = FloatToInt $p2[0]
BPN_CreateImageClassifier bpn ByteImage $size $hidden 0 $nrClasses Bias 0 2!
BPN_TrainCIS bpn cis $nrePOCHS $p2[1] $p2[2]
$evalRes = BPN_EvaluateCIS bpn cis_eval 0.7 LowDetails
$error = GetNthWord 1 $evalRes
$nrMiss = GetNthWord 3 $evalRes
$nrLow = GetNthWord 4 $evalRes
$_cost = $_nrMiss * 1000 + $_nrLow + $_error
return $_cost

```

Fitness criteria

EXECUTE F9 | Abort | Continue | Single step | Step into | Reset

Server ...

28-aug-08 Genetic Algorithms 67

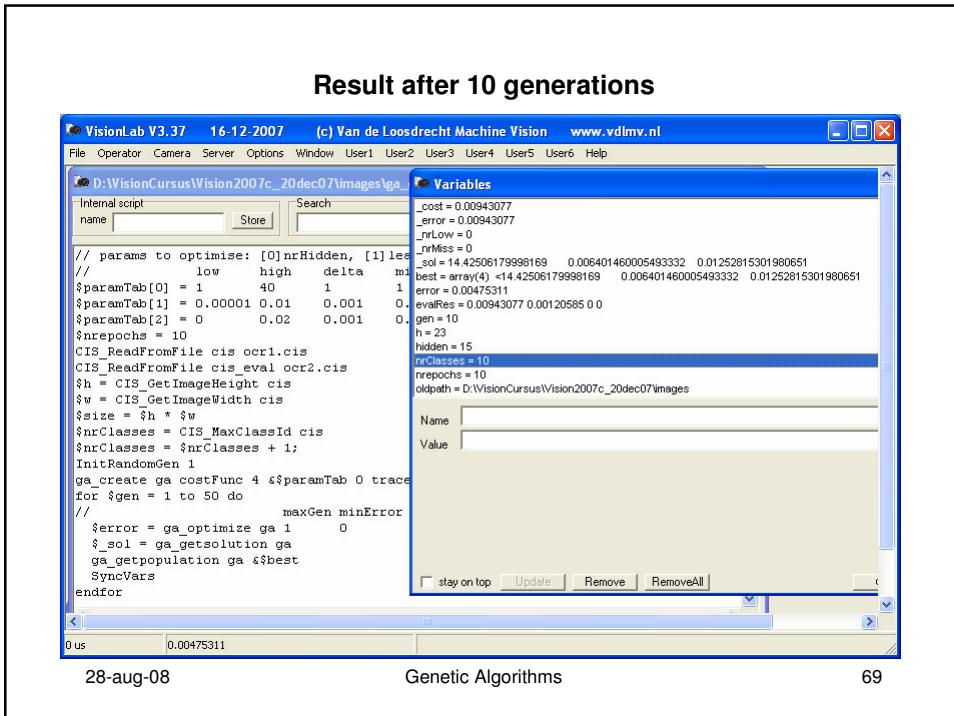
Result after 2 generations

Variable	Value
cost	5.04917
error	0.0491661
nrLow	5
nrMiss	0
_sol	[13.54255806146428, 0.006404246345408491, 0.01253739890743736]
best	[array(4) {13.54255806146428, 0.006404246345408491, 0.01253739890743736}
evalRes	0.0491661 0.00525935 0.5
3.0.697275 0	
3.0.636954 8	
9.9.0.586436 12	
9.9.0.623756 14	
9.9.0.562681 18	
gen	2

Variables

Name | Value | stay on top | Update | Remove | RemoveAll

28-aug-08 Genetic Algorithms 68



Example: Optimising Neural Network selecting features

In this example the neural network with feature vector for recognizing the OCR with feature vector will be optimized from the chapter about Neural Networks

The fitness criteria is chosen as:

- Number of false classifications * 1000 +
- Number of low confidences +
- Error in output layer

Parameters to optimize:

- Which features to select
If param >= 1 then feature is selected

28-aug-08 Genetic Algorithms 70

Demonstration Optimising Neural Network selecting features

- Open scripts:
 - gen_ga_cfs.jls script for generating CFS
 - ga_features.jls: the main program
 - ga_featuresCostFunc.jls: calculation the fitness of a solution
- Run script gen_ga_cfs.jls to generate ga.cfs
- Open variable window
- Run script ga_features.jls
- Note 1: this is time consuming, for real optimisation \$nrEpochs and population size should be greater
- Note 2: if calculations at server side takes much time the client will generate a timeout exception. The timeout value can be changed in the Server Options form in the Server menu.

28-aug-08

Genetic Algorithms

71

gen_ga_cfs.jls: generate ga.cfs

```

VisionLab V3.37 1-1-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help

C:\VisionCursus\Vision2007\c\images\gen_ga_cfs.jls
Internal script name: Store Search Find Line number 68 Indentation + - c
3

$tl = removefirstword &$ana
$t = getnthfromvector 1 $t1
$t1 = getnthfromvector .
$w = removefirstword org (Int16Image Stretch)
$t = $t - 1
$t1 = $t1 - 1
$w = $w + 2
$tl = $tl + 2
roi org roi $t $t1 $w
$c(className = cfs clas
contraststretch roi 0
copy roi new
cfs_addimage cfs new
display roi
endfor $i

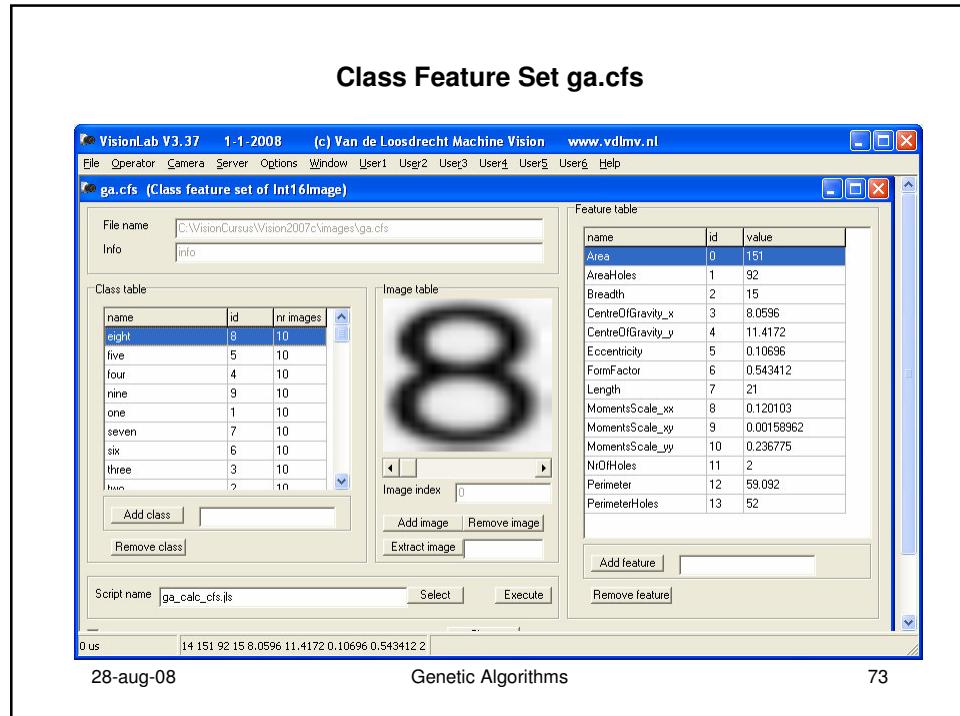
Execute F9 Abort Ctrl+Y Protecte

```

28-aug-08

Genetic Algorithms

72



ga_features.jls

```

Internal script
name | Store | Search | Find | Line number | Indentation
-----|-----|-----|-----|-----|-----|
$AddScript costFunc ga_featuresCostFunc.jls
InitRandomGen 1
CFS_ReadFromFile cfs ga.cfs
$nrEpochs = 10
$learnrate = 0.005
$momentum = 0.01
$nr_hidden = 20
$maxIndex = 13
$pTab[0] = Area
$pTab[1] = AreaHoles
$pTab[2] = Breadth
$pTab[3] = CG.x
$pTab[4] = CG.y
$pTab[5] = Ecc
$pTab[6] = FF
$pTab[7] = "Length"
$pTab[8] = MSxx
$pTab[9] = MSxy
$pTab[10] = MSyy
$pTab[11] = NrOfHoles
$pTab[12] = Perimeter
$pTab[13] = PerHoles
//          low   high   delta   micro   mutationP deltaP

```

\$pTab is table used for displaying result

28-aug-08 Genetic Algorithms 74

ga_features.jls

```

VisionLab V3.37 1-1-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2007c\images\ga_features.jls
Internal script Search Find Line number Jump Indentation
name [ ] Store [ ] [ ] [ ] [ ] [ ] [ ]
$paramTab[13] = PerHoles
// low high delta micro mutationP deltaP
$paramTab[0] = 0 2 1 1 0.2 0.5
for $i = 1 to $maxIndex do
    $paramTab[$i] = $paramTab[0]
endfor
ga_create ga_costFunc 10 &$paramTab 10 tracefile.txt
for $gen = 1 to 50 do
    // maxGen minError deltaError microP
    $error = ga_optimize ga 1 0 0 0.9
    $sol = ga_getsolution ga
    VarToarray &$sol &$tab
    $sol = ""
    for $i = 0 to $maxIndex do
        if $tab[$i] >= 1 then
            $sol = Concat $sol $paramTab[$i]
        endif
    endfor
    SyncVars
endfor

```

Convert solution to text string

0 us 14 151 92 15 8.0596 11.4172 0.10696 0.5434122

28-aug-08 Genetic Algorithms 75

ga_featuresCostFunc.jls

```

VisionLab V3.37 26-1-2008 (c) Van de Loosdrecht Machine Vision www.vdlmv.nl
File Operator Camera Server Options Window User1 User2 User3 User4 User5 User6 Help
C:\VisionCursus\Vision2008c\new_GA\ga_featuresCostFunc.jls
Internal script Search Find Line number Jump Indentation
name [ ] Store [ ] [ ] [ ] [ ] [ ] [ ]
// ga example feature selection
// Jaap van de Loosdrecht, 26-2-2008

$select = ""
for $j = 0 to $maxIndex do
    if $p2[$j] >= 1 then
        $select = $select . 1
    else
        $select = $select . 0
    endif
endfor
CFS_SetSelectTab cfs $select
bpn_createfeatureclassifier bpn cfs $nr_hidden 0 Bias
bpn_traincfs bpn cfs $nrEpochs $learnrate $momentum
$evalRes = bpn_evaluatecfs bpn cfs 0.7 LowDetails
$error = GetNthWord 1 $evalRes
$nrMiss = GetNthWord 3 $evalRes
$nrLow = GetNthWord 4 $evalRes
$cost = $nrMiss * 1000 + $nrLow + $error
return $cost

```

Fitness criteria

[Execute F9] Abort Ctrl+V Continue Single step Step into Reset

0 us

28-aug-08 Genetic Algorithms 76

