



Computer Vision

Genetic Algorithms

10 April 2018

Copyright © 2001 – 2018 by
NHL Stenden Hogeschool and Van de Loosdrecht Machine Vision BV
All rights reserved

j.van.de.loosdrecht@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

27-aug-18

Genetic Algorithms

2

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 chances to survive and produce adapted offspring.

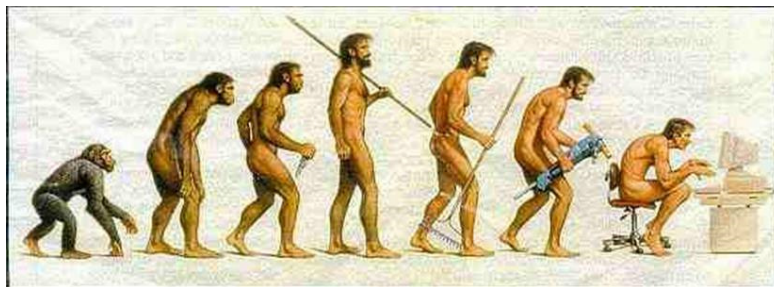
In short: “survival of the fittest”

27-aug-18

Genetic Algorithms

3

Introduction Genetic Algorithms



27-aug-18

Genetic Algorithms

4

Example Mother Nature

DNA represent the blueprint of a being

DNA mother: 11011010111010101001

DNA father: 10011010110001001011

Mating, DNA child:

- Crossover: 11011010110001001011
- Mutation: 10011110110101001011

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

27-aug-18

Genetic Algorithms

5

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

27-aug-18

Genetic Algorithms

6

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

27-aug-18

Genetic Algorithms

7

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

27-aug-18

Genetic Algorithms

8

Demonstration: finding best fit for a polynomial

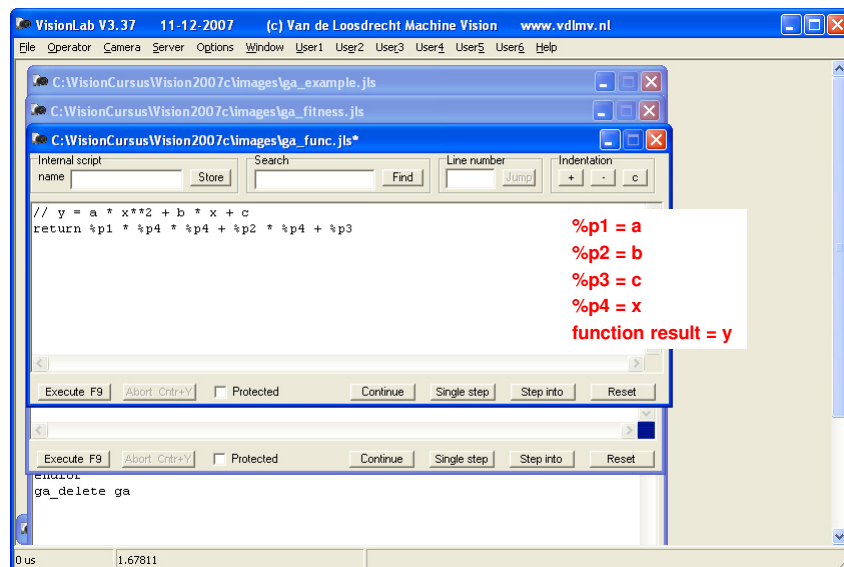
- Open scripts:
 - `ga_func.jls`: calculation of the polynomial
 - `ga_example.jls`: the main program
 - `ga_fitness.jls`: calculation the fitness of a solution
- Open variable window
- Run script `ga_example.jls`
- Change line to create ga in:
`ga_create ga ga_fitness 10 &$paramTab 10 tracefile.txt`
- 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

27-aug-18

Genetic Algorithms

9

ga_func.jls



27-aug-18

Genetic Algorithms

10

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

```

Generate point cloud
a = 2, b = 3, c = 4

Initialize parameter
description table

27-aug-18

Genetic Algorithms

11

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

```

Create ga with
paramTab and fitness

27-aug-18

Genetic Algorithms

12

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

```

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

27-aug-18

Genetic Algorithms

13

ga_fitness.jls

```

// script: ga_fitness.jls
// optimising polynomial with GA
// Jaap van de Loosdrecht, 26-2-2008

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

```

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

27-aug-18

Genetic Algorithms

14

Run script ga_example

27-aug-18

Genetic Algorithms

15

tracefile.txt

27-aug-18

Genetic Algorithms

16

Example: Camera calibration

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

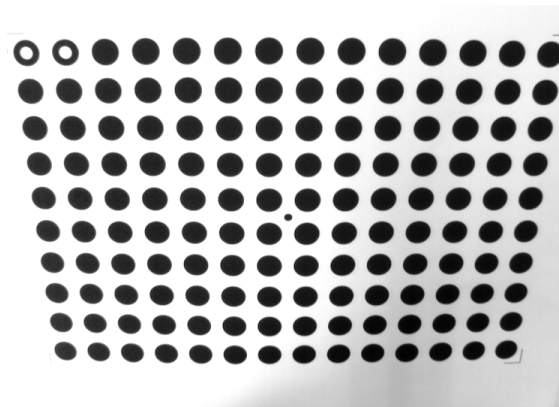
27-aug-18

Genetic Algorithms

17

Calibration Pattern

A fixed pattern of circles is used to calibrate the camera



27-aug-18

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.

27-aug-18

Genetic Algorithms

19

Demonstration Camera Calibration

27-aug-18

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

27-aug-18

Genetic Algorithms

21

GA_Create

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

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

27-aug-18

Genetic Algorithms

22

GA_Create

Parameters continued:

- **paramTab**: array with the description of the parameters:
 - **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]

27-aug-18

Genetic Algorithms

23

GA_CreateAll

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

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

27-aug-18

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

27-aug-18

Genetic Algorithms

25

GA_CreateAll

Parameters continued:

- **paramTab:** array with the description of the parameters:
 - **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]

27-aug-18

Genetic Algorithms

26

GA_Delete

GA_Delete (gaName)

Delete an instance of a Genetic Algorithm

Parameter:

- **gaName:** name of variable

27-aug-18

Genetic Algorithms

27

GA_RandomInitialize

GA_RandomInitialize (gaName, populationSize)

Random initialise the population

Parameters:

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

27-aug-18

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

27-aug-18

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

27-aug-18

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

27-aug-18

Genetic Algorithms

31

GA_GetSolution

GA_GetSolution (gaName)

Function result is the best solution found

Parameter:

- **gaName:** name of variable

27-aug-18

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

27-aug-18

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`

27-aug-18

Genetic Algorithms

34

ga_all_example.jls

```

laddScript ga_func ga_func.jls
laddScript ga_fitness ga_fitness_all.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_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
endfor
  
```

Use ga_fitness_all.jls as fitness function

Use ga_createall instead of ga_create

27-aug-18 Genetic Algorithms 35

ga_fitness_all.jls

```

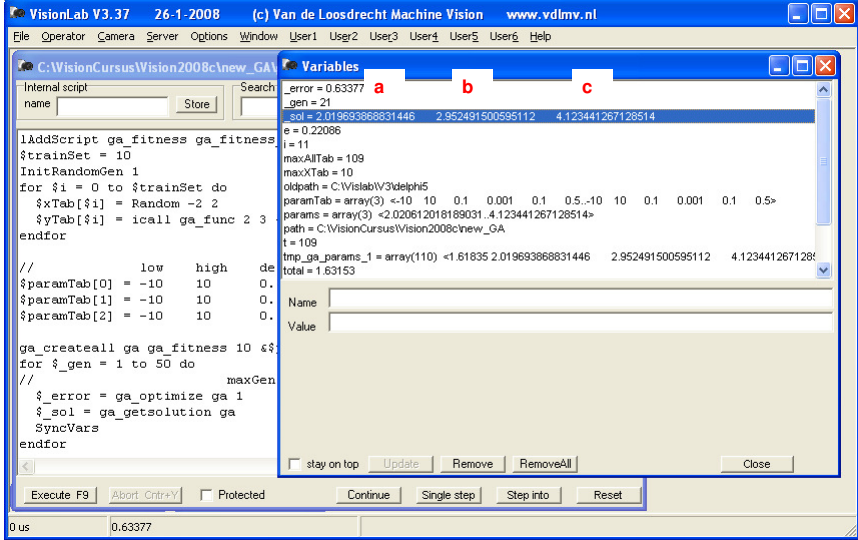
for $t = 0 to $maxAllTab do
  $total = 0
  VarToArray $p2[$t] $params
  for $i = 0 to $maxXTab do
    $y_ga = icall ga_func $params[0] $params[1] $params[2] $:
    $e = $yTab[$i] - $y_ga
    $e = Fabs $e
    $total = $total + $e
    if $total > %p1 then
      $i = $maxXTab + 1
    endif
  endfor
  $p2[$t]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

Calculate the total error of the point cloud
 %p1 = cutoff error
 %p2 = array containing the parameters for each being to optimize.
 \$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.

27-aug-18 Genetic Algorithms 36

Run script ga_all_example



The screenshot shows the VisionLab V3.37 interface. The main window displays a script titled "C:\VisionCursus\Vision2008\new_GA". The script is a genetic algorithm implementation for a Travelling Salesman Problem. The script includes initialization, fitness calculation, and optimization steps. The "Variables" window on the right shows the current state of the algorithm:

- _error** = 0.63377
- _gen** = 21
- _sol** = 2.019693868831446 2.952491500595112 4.123441267128514
- e** = 0.22086
- i** = 11
- maxAllTab** = 109
- maxXTab** = 10
- oldpath** = C:\VisionLab\3\delphi5
- paramTab** = array(3) <-10 10 0.1 0.001 0.1 0.5...10 10 0.1 0.001 0.1 0.5>
- params** = array(3) <2.020612018189031 4.123441267128514>
- path** = C:\VisionCursus\Vision2008\new_GA
- t** = 109
- tmp_ga_params_1** = array(110) <1.61835 2.019693868831446 2.952491500595112 4.123441267128514>
- total** = 1.63153

The script execution window shows the following code:

```

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

//          low      high    de
$paramTab[0] = -10    10     0.
$paramTab[1] = -10    10     0.
$paramTab[2] = -10    10     0.

ga_createall ga ga_fitness 10 $
for $gen = 1 to 50 do
  //          maxGen
  $error = ga_optimize ga 1
  $sol = ga_getsolution ga
  SyncVars
endfor
  
```

At the bottom of the script window, the execution time is shown as 0 us and the current fitness value is 0.63377.

27-aug-18 Genetic Algorithms 37

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 indication 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 indication 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

27-aug-18

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

27-aug-18

Genetic Algorithms

40

ga_travel.jls (1)

The screenshot shows the VisionLab V3.37 interface. The script editor displays the following code:

```
// create image with cities
create img Int16Image 255 255
setallpixels img 0
noise img 0.02 1 1
addborder img img 20 20 20 20 0
$cities = labelblobs img EightConnected
blobanalysisarray img $ba_arr NoSort CentreOfGravity UseX CentreOfGravity
copy img imgcities
clippixelvalue imgcities 0 1
Dilation imgcities imgcities 5 5 2 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0
$count = $cities - 1

RemoveVar $x_src
RemoveVar $y_src
// find coordinates of cities
for $i = 0 to $count
  $var = $ba_arr[$i]
  $label = getnthword 1 $var
  $cog = getnthword 2 $var
  $x = getnthfromvector 1 $cog
  $y = getnthfromvector 2 $cog
```

A red annotation "Generate image with random cities" is placed over the code. The status bar at the bottom shows "0 us".

27-aug-18

Genetic Algorithms

41

ga_travel.jls (2)

The screenshot shows the VisionLab V3.37 interface. The script editor displays the following code:

```
copy img imgcities
clippixelvalue imgcities 0 1
Dilation imgcities imgcities 5 5 2 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0
$count = $cities - 1

RemoveVar $x_src
RemoveVar $y_src
// find coordinates of cities
for $i = 0 to $count
  $var = $ba_arr[$i]
  $label = getnthword 1 $var
  $cog = getnthword 2 $var
  $x = getnthfromvector 1 $cog
  $y = getnthfromvector 2 $cog
  $x_src[$i] = $x
  $y_src[$i] = $y
endfor

laddscript ga_travel_costfunc ga_travel_costfunc.jls
laddscript ga_travel_route ga_travel_route.jls
RemoveVar $paramtab
// <low> <high> <delta> <micro> <mutationP> <deltaP>
```

A red annotation explains the arrays: "\$x_src: array with x-coordinate of cities" and "\$y_src: array with y-coordinate of cities". The status bar at the bottom shows "0 us" and "1329.82".

27-aug-18

Genetic Algorithms

42

ga_travel.jls (3)

```

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
    $i1 = $i - 1
    $c_src = vector $$x[$i1] $$y[$i1]
    $c_dst = vector $$x[$i] $$y[$i]
  
```

Initialize parameter description table

Generate n generations

27-aug-18

Genetic Algorithms

43

ga_travel.jls (4)

```

  $$dist = GA Optimize ga 1 0 0 0.5
  $sol = GA GetSolution ga
  vartoarray $$sol $$sol_arr
  icall ga_travel_route $sol_arr $x_src $y_src $$x $$y
  // display result
  copy img img_disp
  setallpixels img_disp 0
  $count = $cities - 1
  for $i = 1 to $count
    $i1 = $i - 1
    $c_src = vector $$x[$i1] $$y[$i1]
    $c_dst = vector $$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

```

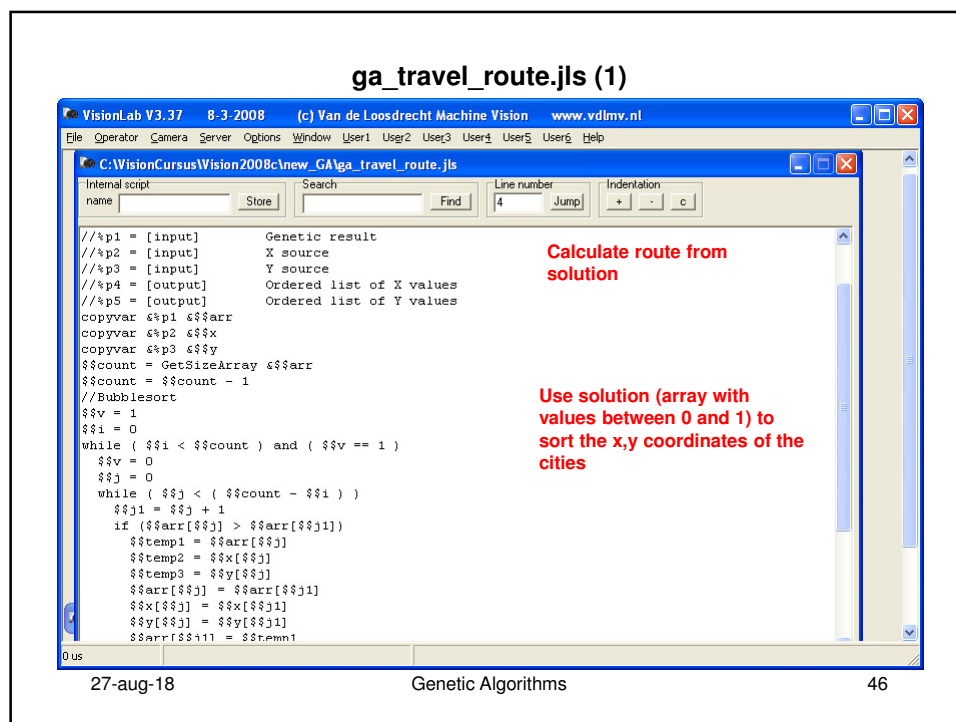
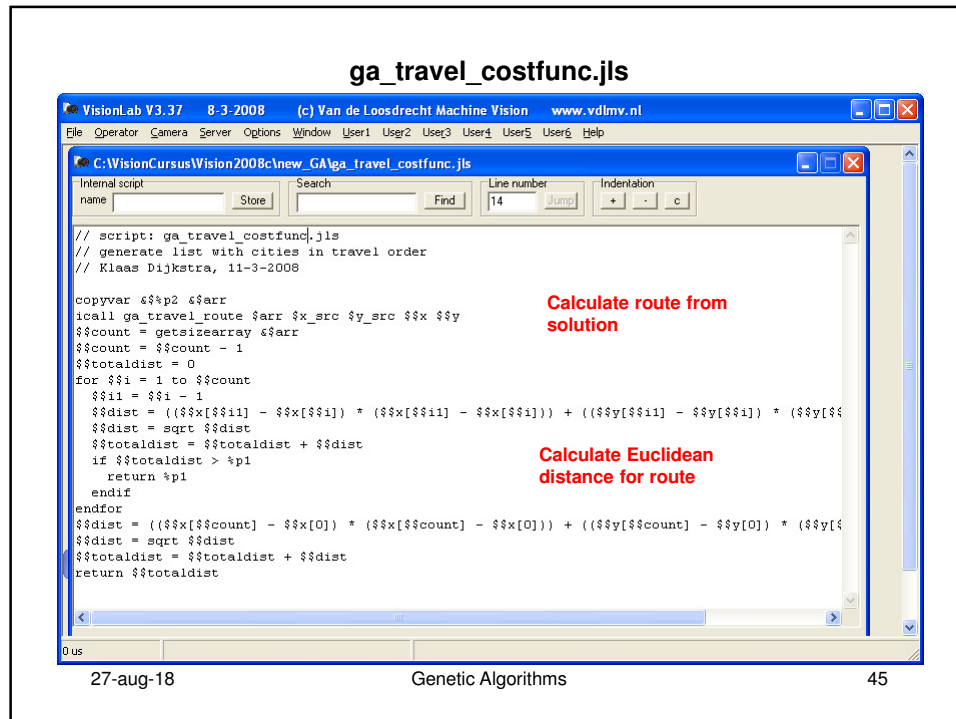
Generate route from GA solution

Display route in image

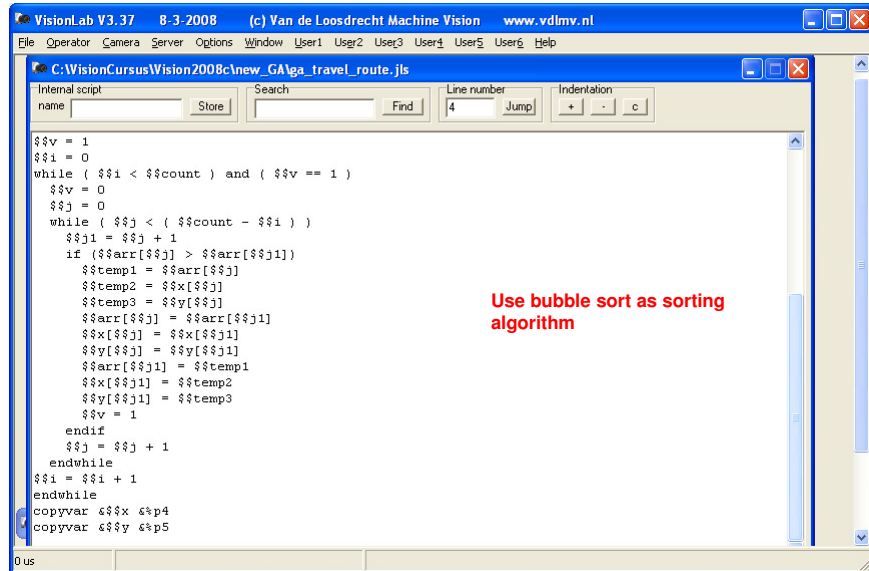
27-aug-18

Genetic Algorithms

44



ga_travel_route.jls (2)



```

VisionLab V3.37  8-3-2008  (c) Van de Loosdrecht Machine Vision  www.vdlnv.nl
File  Operator  Camera  Server  Options  Window  User1  User2  User3  User4  User5  User6  Help

C:\VisionCursus\Vision2008\new_GA\ga_travel_route.jls

Internal script
name  Store  Search  Find  Line number  Indentation
4  Jump  +  -  c

$$v = 1
$$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[$$j]
      $$temp2 = $$x[$$j]
      $$temp3 = $$y[$$j]
      $$arr[$$j] = $$arr[$$j1]
      $$x[$$j] = $$x[$$j1]
      $$y[$$j] = $$y[$$j1]
      $$arr[$$j1] = $$temp1
      $$x[$$j1] = $$temp2
      $$y[$$j1] = $$temp3
    endif
    $$j = $$j + 1
  endwhile
  $$i = $$i + 1
endwhile
copyvar $$x 4tp4
copyvar $$y 4tp5
0 us

```

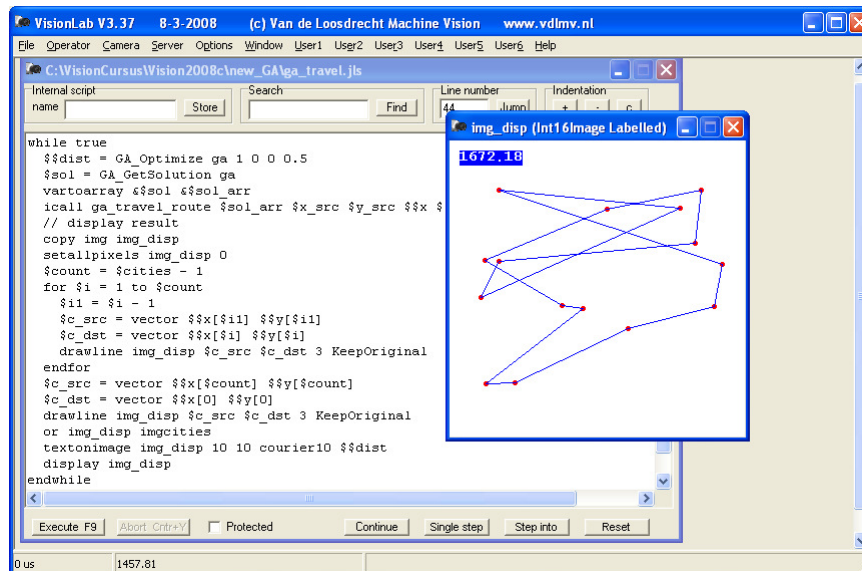
Use bubble sort as sorting algorithm

27-aug-18

Genetic Algorithms

47

Running TSP (1)



```

VisionLab V3.37  8-3-2008  (c) Van de Loosdrecht Machine Vision  www.vdlnv.nl
File  Operator  Camera  Server  Options  Window  User1  User2  User3  User4  User5  User6  Help

C:\VisionCursus\Vision2008\new_GA\ga_travel_route.jls

while true
  $$dist = GA_Optimize ga 1 0 0 0.5
  $sol = GA_GetSolution ga
  vartoarray c$sol c$sol_arr
  icall ga_travel_route $sol_arr $x_src $y_src $$x $
  // display result
  copy img img_disp
  setallpixels img_disp 0
  $count = $cities - 1
  for $i = 1 to $count
    $i1 = $i - 1
    $c_src = vector $$x[$i1] $$y[$i1]
    $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
0 us
1457.81

```

img_disp (Int16Image Labelled)

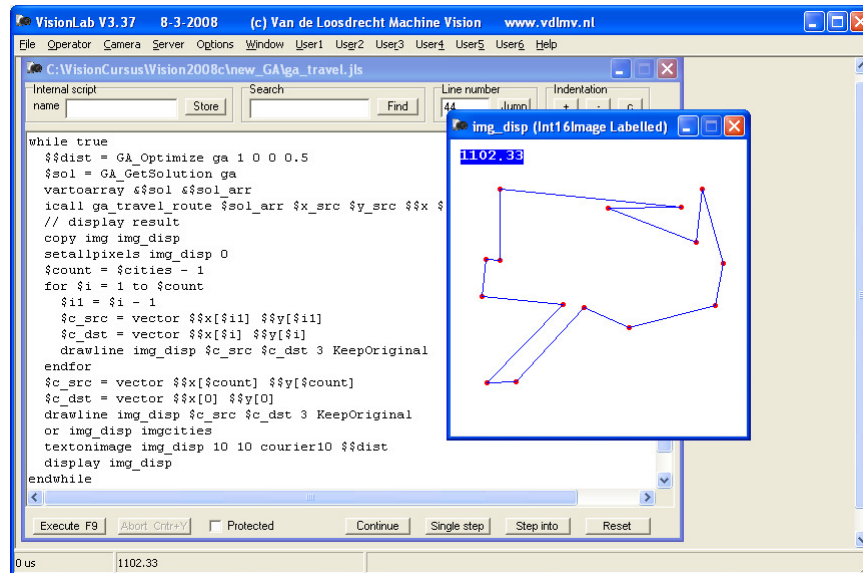
1672.18

27-aug-18

Genetic Algorithms

48

Running TSP (2)

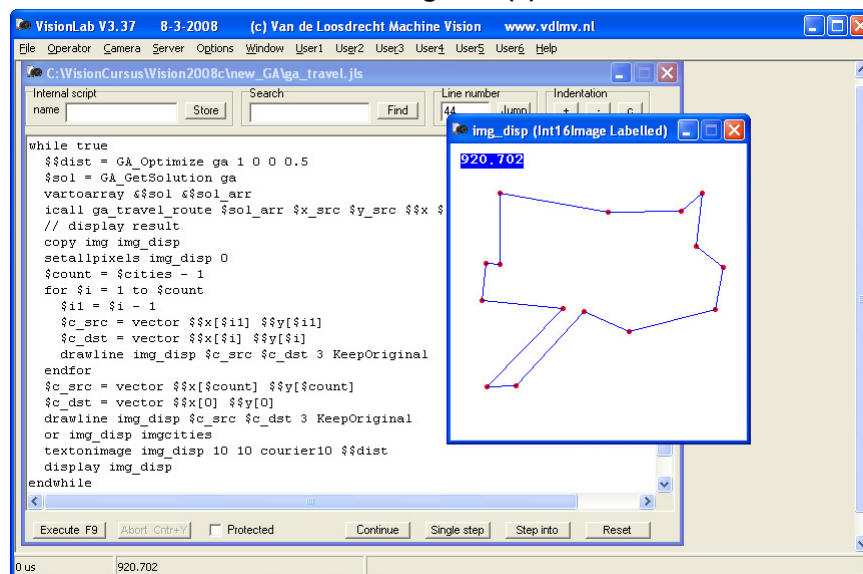


27-aug-18

Genetic Algorithms

49

Running TSP (3)



27-aug-18

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 indication if item is in KnapSack
 - **Fitness function:**
total value of items in KnapSack not exceeding max weight

27-aug-18

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 >= 1)**

27-aug-18

Genetic Algorithms

52

Example: KnapSack

- **Fitness function:**
total value of items in KnapSack not exceeding max weight
 - **Fitness function:**
if total weight \leq 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 \leq max weight then
topValue - total value of items in KnapSack
else
topValue
 - **Where topValue > sum value of all items**

27-aug-18

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

27-aug-18

Genetic Algorithms

54

knapsack.jls

```

lAddScript costFunc knapsackCostFunc.jls
$maxWeight = 15
$topValue = 1000
$nrItems = 20
$maxIndex = $nrItems - 1
InitRandomGen 1
for $i = 0 to $maxIndex do
    $wTab[$i] = Random 1 5
    $vTab[$i] = Random 1 5
endfor
//
$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
    //
    $error = ga_optimize ga 1    0    0    0.9
    $maxValue = $topValue - $error
    $sol = ga_getsolution ga
    VarToArray $sol $sTab
    $sack = " "
    $weight = 0
    for $i = 0 to $maxIndex do
        //
        _maxValue = 30.97 _sack = 110101001001

```

27-aug-18 Genetic Algorithms 55

knapsack.jls

```

    $wTab[$i] = Random 1 5
endfor
//
$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
    //
    $error = ga_optimize ga 1    0    0    0.9
    $maxValue = $topValue - $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

```

27-aug-18 Genetic Algorithms 56

knapsackCostFunc.jls

```

// script: knapsackCostFunc.jls
// optimising knapsack with GA
// Jaap van de Loosdrecht, 26-2-2008

$pe
sol
cwo
$value = 0
$weight = 0
for $j = 0 to $maxIndex do
  $me if $p2[$j] >= 1 then
    $to $weight = $weight + $vTab[$j]
    $nr if $weight > $maxWeight then
      $me return $stopValue
    endnr
  endfor
  $value = $value + $vTab[$j]
endfor
return $stopValue - $value

```

\$p2[0] = first item
 \$p2[19] = last item
 %p1 (cutoff error) is not used
 Sum weight of items in sack
 Reverse cost function

27-aug-18 Genetic Algorithms 57

Run knapsack.jls

```

// low high delta micro mutationP deltaP
$paramTab[0] = 0 2 1 0 0 0
for $i = 1 to $maxIndex do
  $paramTab[$i] = $paramTab[0]
endfor
ga_create ga costFunc 10 $paramTab
for $gen = 1 to 50 do
  // maxGen
  $error = ga_optimize ga 1
  $_maxValue = $topValue - $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 + $vTab[$i]
    else
      $s = 0
    endif
    $_sack = Concat $_sack $s
  endfor
  SyncVars
endfor

```

Variables

- _maxValue = 30.97
- _sack = 11010100100010011100
- _weight = 14.9734
- error = 969.03
- gen = 15
- i = 19
- ITab = array(20) <1.87160863063448 0.833307901242103>
- Items = 1.87160863063448 0.6493423261207922 0.6980498672444838 1.193365276039918 0.151585436567
- j = 16
- maxIndex = 19
- maxWeight = 15
- nrItems = 20
- paramTab = array(20) <0 2 1 1 0.2 0.5 0 2 1 1 0.2 0.5>
- s = 0

Name	vTab
index	value
0	3.25434
1	4.23496
2	2.91949
3	4.58385

27-aug-18 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

27-aug-18

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 \$nepochs and population size should be greater
- Note 2: if calculations at server side takes much time the client will generated a timeout exception. The timeout value can be changed in the Server Options form in the Server menu.

27-aug-18

Genetic Algorithms

60

ga_ocr.jls

```

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

```

Initialize parameter description table
 [0]: nr hidden nodes
 [1]: learnrate
 [2]: momentum

27-aug-18

Genetic Algorithms

61

ga_ocr_CostFunc.jls

```

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

$hidden = FloatToInt $p2[0]
1 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

```

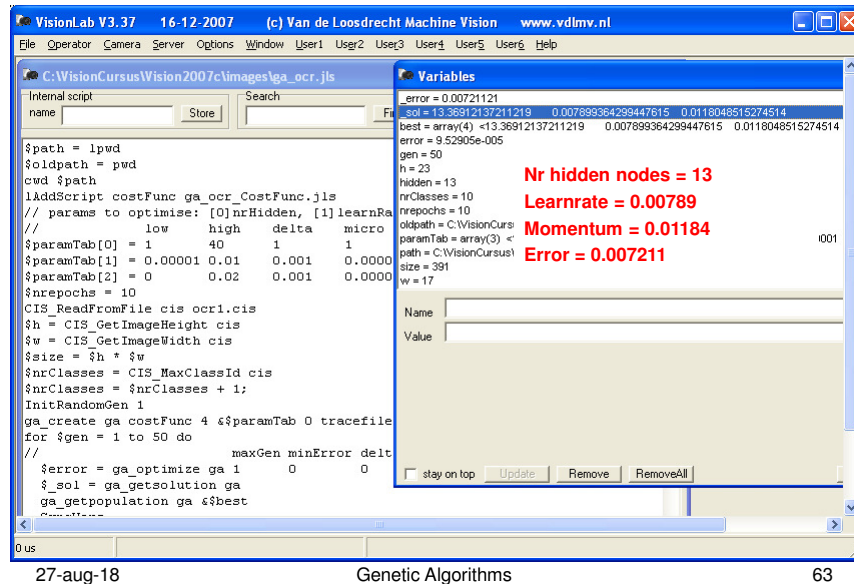
Create and train Neural network
 %p1 = cutoff error, not used
 %p2[0] = nr hidden nodes
 %p2[1] = learnrate
 %p2[2] = momentum
 Function result = training error

27-aug-18

Genetic Algorithms

62

Result of GA optimizing



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 \$nreps should be greater
- Note 2: if calculations at server side takes much time the client will generated a timeout exception. The timeout value can be changed in the Server Options form in the Server menu.

27-aug-18

Genetic Algorithms

65

ga_ocr_eval.jls

```

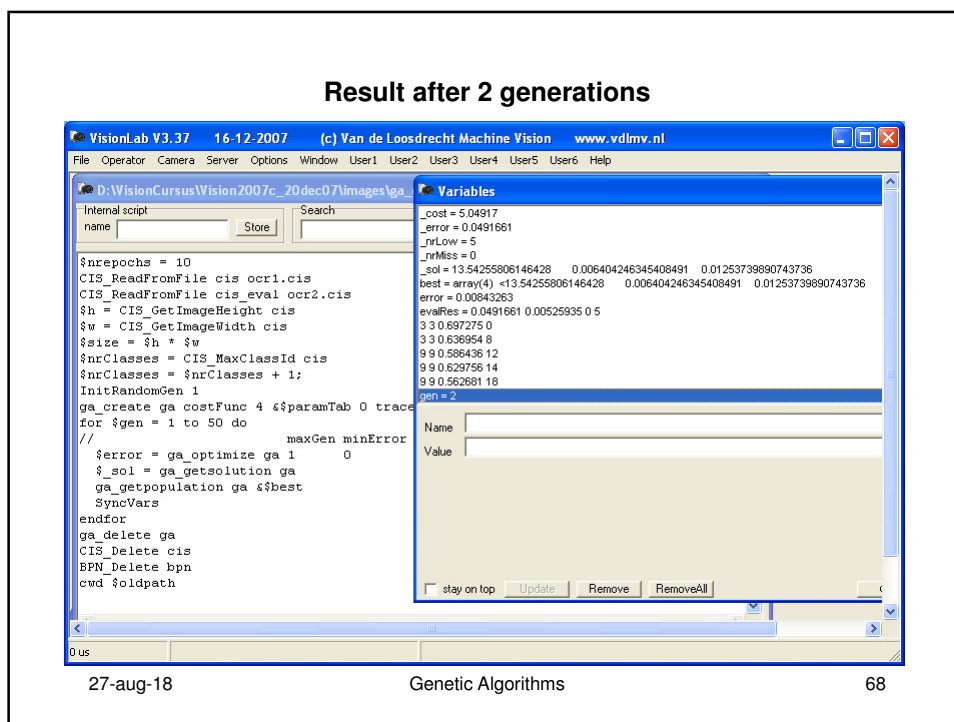
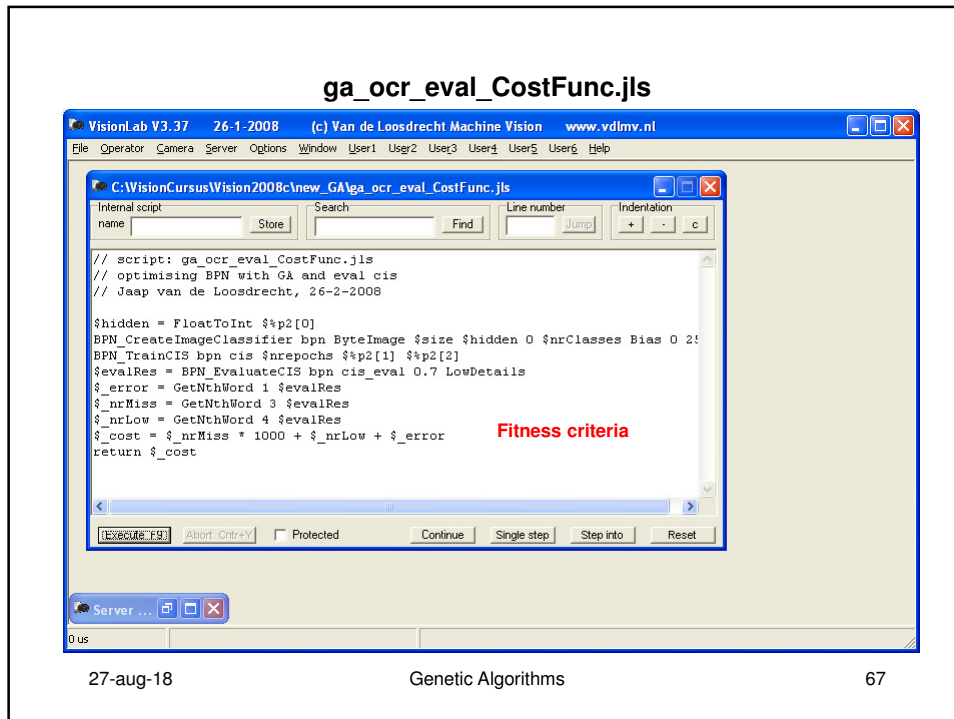
lAddScript costFunc ga_ocr_eval_CostFunc.jls
// params to optimize: [0]nrHidden, [1]learnRate and [2]momentum
//      low      high      delta      micro      mutationP      deltaP
$paramTab[0] = 1      40      1      1      0.2      0.5
$paramTab[1] = 0.00001 0.01    0.001  0.00001  0.2      0.5
$paramTab[2] = 0      0.02    0.001  0.00001  0.2      0.5
$nreps = 10
CIS_ReadFromFile cis ocr1.cis
CIS_ReadFromFile cis_eval ocr2.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

```

27-aug-18

Genetic Algorithms

66



Result after 10 generations

The screenshot shows the VisionLab V3.37 software interface. The main window displays a script with parameters for a neural network. The 'Variables' window shows the current state of the algorithm.

```

// params to optimise: [0]nrHidden, [1]le
//      low      high  delta  m
$paramTab[0] = 1    40    1      1
$paramTab[1] = 0.00001 0.01  0.001 0
$paramTab[2] = 0    0.02  0.001 0
$nrEpochs = 10
CIS_ReadFromFile cis ocr1.cis
CIS_ReadFromFile cis_eval ocr2.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 trace
for $gen = 1 to 50 do
  //      maxGen minError
  $error = ga_optimize ga 1 0
  $sol = ga_getsolution ga
  ga_getpopulation ga $best
  SyncVars
endfor

```

The 'Variables' window shows the following values:

```

_cost = 0.00943077
_error = 0.00943077
_nrLow = 0
_nrMiss = 0
_sol = 14.42506179998169 0.006401460005493332 0.01252815301980651
best = array(4) <14.42506179998169 0.006401460005493332 0.01252815301980651
error = 0.00475311
evalRes = 0.00943077 0.00120585 0 0
gen = 10
h = 23
hidden = 15
nrClasses = 10
nrEpochs = 10
oldpath = D:\VisionCursus\Vision2007c_20dec07\images

```

27-aug-18

Genetic Algorithms

69

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

27-aug-18

Genetic Algorithms

70

Demonstration Optimising Neural Network selecting features

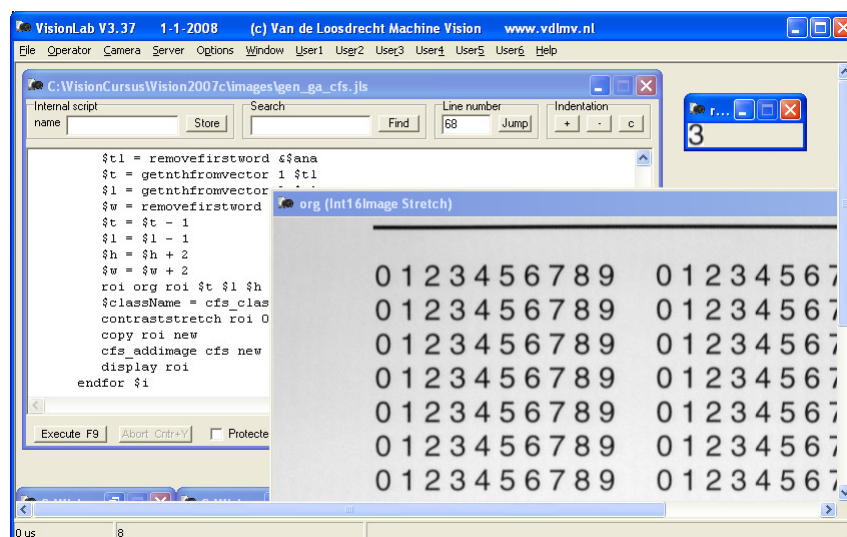
- Open scripts:
 - `gen_ga_cfs.jls` script for generating CFS
 - `ga_features.jls`: the main program
 - `ga_featuresCostFunc.jls`: calculation the fitness of a solution
- Run script `gen_ga_cfs.jls` to generate `ga.cfs`
- Open variable window
- Run script `ga_features.jls`
- Note 1: this is time consuming,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

27-aug-18

Genetic Algorithms

71

gen_ga_cfs.jls: generate ga.cfs



27-aug-18

Genetic Algorithms

72

Class Feature Set ga.cfs

The screenshot shows the VisionLab V3.37 interface with the Class Feature Set ga.cfs loaded. The Class table lists classes and their image counts. The Image table shows a selected image of the digit '8'. The Feature table lists various features and their values.

name	id	nr images
eight	8	10
five	5	10
four	4	10
nine	9	10
one	1	10
seven	7	10
six	6	10
three	3	10
two	2	10

name	id	value
Area	0	151
AreaHoles	1	92
Breadth	2	15
CentreOfGravity_x	3	8.0596
CentreOfGravity_y	4	11.4172
Eccentricity	5	0.10696
FormFactor	6	0.543412
Length	7	21
MomentsScale_xx	8	0.120103
MomentsScale_xy	9	0.00158962
MomentsScale_yy	10	0.236775
NrOfHoles	11	2
Perimeter	12	59.092
PerimeterHoles	13	52

27-aug-18

Genetic Algorithms

73

ga_features.jls

The screenshot shows the VisionLab V3.37 interface with the ga_features.jls script loaded. The script defines a cost function and initializes a genetic algorithm. A red text annotation points to the \$pTab variable.

```

lAddScript 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

27-aug-18

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 name Store Search Find Line number Indentation
41 Jump + - c

$pTab[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 &$sTab
  $sol = " "
  for $i = 0 to $maxIndex do
    if $sTab[$i] >= 1 then
      $sol = Concat $sol $pTab[$i]
    endif
  endfor
  SyncVars
endfor
0 us 14 151 92 15 8.0596 11.4172 0.10696 0.543412 2

```

Convert solution to text string

27-aug-18

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 name Store Search Find Line number Indentation

// 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
bnp_createfeatureclassifier bnp cfs $nr_hidden 0 Bias
bnp_traincfs bnp cfs $nrEpochs $learnrate $momentum
$evalRes = bnp_evaluatecfs bnp cfs 0.7 LowDetails
$error = GetNthWord 1 $evalRes
$nrMiss = GetNthWord 3 $evalRes
$nrLow = GetNthWord 4 $evalRes
$cost = $nrMiss * 1000 + $nrLow + $error
return $cost
0 us

```

\$%p2[0] = first param
 \$%p2[1] = second param
 etc
 If param >= 1 then select feature

Fitness criteria

27-aug-18

Genetic Algorithms

76

