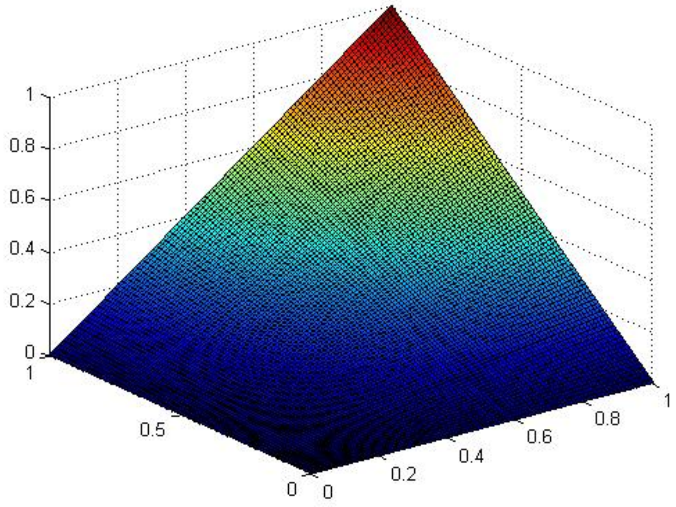
Feed Forward Neural Network with Genetic Algorithm

Problem definition and training data selection

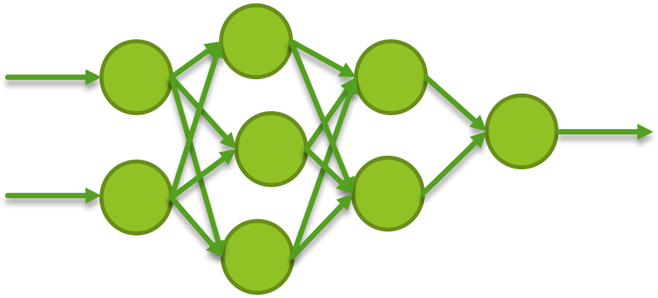
- Z = f(x,y) = x*y
- 0<=x<=1
- 0<=y<=1



- x and y varied from 0 to 1 in intervals of 0.01
- z = f(x,y) will have 100x100 grid points
- 100 points randomly chosen for x and y each
- This gives z = f(x,y) as 100 points for training

Neural Network Structure

- 2 inputs – x and y
- 2 hidden layers
  - 3 neurons in first hidden layer
  - 2 neurons in second hidden layer
- 1 output – z = f(x,y)



- 14 edges, each associated with a weight
- y1 = g(w(1)*x + w(2)*y)
- y2 = g(w(3)*x + w(4)*y)
- y3 = g(w(5)*x + w(6)*y)
- z1 = g(w(7)*y1 + w(8)*y2 + w(9)*y3)
- z2 = g(w(10)*y1 + w(11)*y2 + w(12)*y3)
- output = g(w(14)*z1 + w(14)*z2)
- Here, g is chosen to be a sigmoid function
- This neural network encoded in compute_neural.m

Encoding the chromosomes for GA

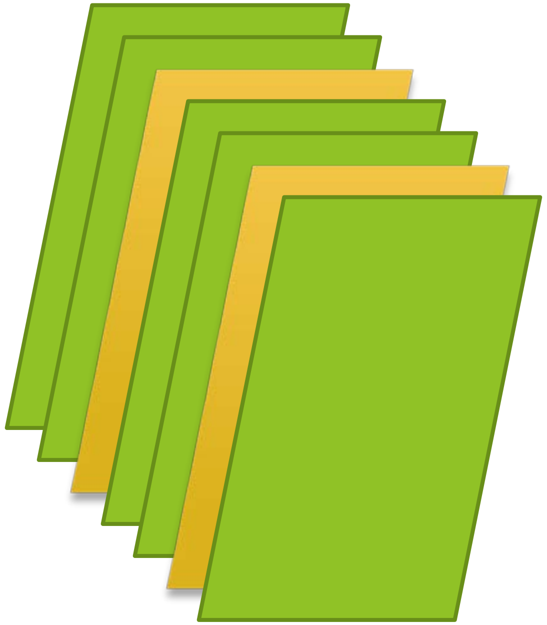
- -6.35<=weight<=6.4
- (-6.35,6.4) = (-127/20,128/20)
- -127 to 128 -> 255 values
- 8 bit binary representation
- 128 added to each weight, so that all are non zero, and then the binary equivalent is filled in an 8 bit array
- Functionality captured by decimal_to_binary.m and binary_to_decimal.m
- 14 edges in the network => 14 weights
- 14 binary arrays needed for the neural network
- A 14x8 matrix created to represent all the weights in the network
- Every such 14x8 matrix represents a chromosome for the GA

GA Formulation

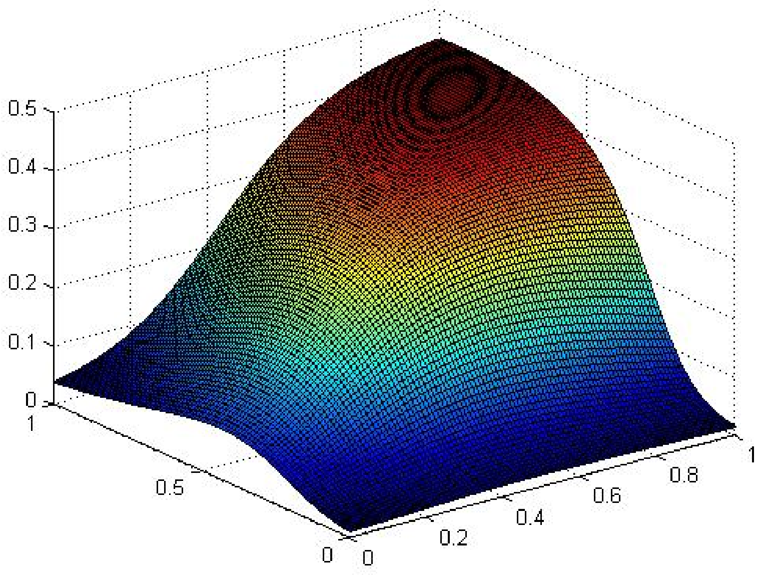
- Random initial population generated by using round(rand(14,8,41))
- 41 is the population size
- Every 14x8 matrix is a chromosome
- Fitness function
  - For a given chromosome and x and y, compute the rms error for (actually taken max error in this case)
- Fitness proportionate selection
  - An array generated with indices of the chromosomes; lower the error, more indices of that chromosome
  - Consider [0.1 0.4 0.3 0.2]
  - The array A generated is similar to [1 2 2 2 2 3 3 3 4 4]
- Crossover
  - Randomly choose indices for the fitness proportionate selection array
  - Corresponding indices are the indices for 2 parents
  - Randomly choose columns from each and exchange
- Mutation
  - randi(1,100) < 10 (10% probability)
- A slight modification is made for better convergence, best of each generation is kept in every new generation
- Capture in generate_offspring.m

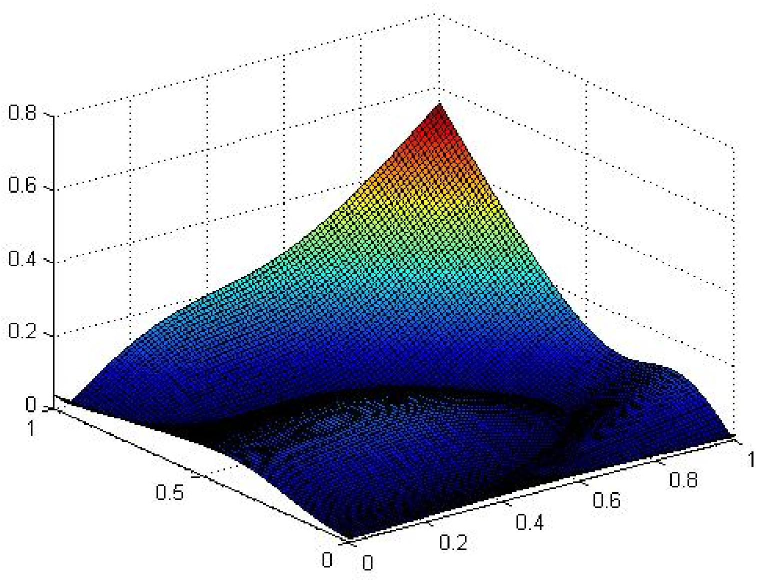
Graphic visualization

```
0 1 0 0 1 0 1 0
1 1 1 0 0 1 1 0
0 1 0 1 1 1 0 1
1 1 1 0 0 0 0 1
1 0 1 0 1 1 1 1
0 1 0 0 1 0 1 1
1 0 1 1 1 0 1 0
0 0 0 0 1 0 1 0
1 0 0 0 1 1 1 1
1 1 1 0 0 0 1 0
0 1 0 1 0 0 1 0
1 1 0 1 0 1 0 0
0 1 1 0 0 1 1 0
1 0 0 0 1 1 0 0
```



Neural Network fitted values, error





Possible manipulations

- Population size (41)
- Range of weights (-6.35 to 6.4)
- Number of neurons (3,2)
- Number of hidden layers (2)
- Resolution of weights (1/20)
- Number of iterations of GA (200)
- Selection of fitness function (max(errors)) (conservative)
- Selection of activation function (sigmoid)