# Encoding

Consider a BPNN network with a configuration 2-2-2 (2 neurons in input layer, 2 neurons in hidden layer and 2 neurons in output layer). The number of weights to be determined are (2 X 2) + (2 X 2) = 8.

Input Neurons

Hidden Neurons

Output Neurons

Inputs

Outputs

W11

W12

W21

V11

W22

V22

V21

V12

Input Layer

Hidden Layer

Output Layer

With each weight being a real number and assuming the number of digits "d" to be randomly generated for representing a weight value as 5, the string "S" representing the chromosome of weights is 8 X 5 = 40 chromosomes in length for every weight. Let represent the 40 chromosomes of P0 (INITIAL population)

Chromosome

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Gene | Gene | Gene | Gene | Gene | Gene | Gene | Gene |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 84321 | 46234 | 78901 | 3210 | 42689 | 63421 | 46421 | 87640 |
|  | | | | | | | |

Choose population size p=40 i.e. generate 40 chromosomes.

# Weight Extraction

Example:

Gene 0: 84321: k=0, d=5, and which is is such that

Hence the weight extracted is:

Similarly for

Gene 1: 46234 yields w1 = –6.234

Gene 2: 78901 yields w2 = +8.901

Gene 3: 32104 yields w3 = –2.104

Gene 4: 42689 yields w4 = –2.689

Gene 5: 63421 yields w5 = +3.421

Gene 6: 46421 yields w6 = –6.421 and

Gene 7: 87640 yields w7 = +7.640

Let be the weight sets extracted from each of by using the above mentioned equation.

# Fitness Function:

Let represent a set of input-output pairs for a problem P to be solved by the BPNN.

Let be the calculated outputs.

Now compute

The root mean square of the error is

And the fitness F1 of the chromosome will be:

# Reproduction

For the first iteration, this step will be ignored along with step of fitness. However, for subsequent steps, we will be having pool of chromosomes and have corresponding fitness of every chromosome. For the ease, we will simply drop chromosomes with least fitness and replace them with duplicate copy of chromosomes having maximum fitness. e.g. are the top 5 most fit chromosomes and are the least fit chromosomes so we will replace with another copy of and with another copy of and with another copy of and with another copy of and with another copy of . The new pool of chromosomes will still have 40 chromosomes including two chromosomes of each.

# Crossover (2-Point crossover, in this case)

In crossover, use random number to choose two random chromosomes from the mating pool. Similarly use random number twice to select 2 cross-sites for crossover. The pair chosen for crossover will be taken away from the mating pool.

# Mutation

Similarly, after crossover, randomly choose a chromosome for mutation. Use random number to decide which part to be mutate. In case of integers several mutation techniques can be adopted like

Remember length of chromosome is 40 and mapping it to hundred means 1:2.5. For integers we may adopt the technique (Option 1) to swap 0 with 5, 1 with 6, 2 with 7, 3 with 8 and 4 with 9 OR (Option 2) we may swap 0 with 9, 1 with 8, 2 with 7, 3 with 6 and 4 with 5 OR (Option 3) we may convert the digit to binary numbers, swap 0 with 1 and 1 with zero and convert it again to decimal number.

# Convergence

Now we have a new pool of chromosomes. Extract the weights from here and use them in neural network. Get the error again and repeat the above mentioned process of getting fitness, selection/reproduction, crossover & mutation till the time 95% chromosomes in the population converge to the same fitness value

# Problem Statement

Design a **GA optimized BPNN system** having user defined number of neurons in the input and output layer, and number of neurons in hidden layer same as in input layer.

1. Inputs from user
   1. Training Data
   2. How many inputs?
   3. How many out puts?
   4. How many datasets (input out pairs)
   5. Get data set
   6. Number of chromosomes to be replaced with best fit ones during reproduction?
   7. Number of Crossover points?
   8. Crossover percentage
   9. Mutation method (from above mentioned available options)
   10. Mutation percentage
   11. Acceptable convergence %age
2. Output
   1. Two weights matrices
   2. Number of iterations performed to achieve required convergence