

NAS using Genetic Algorithm

1st Mayank Gupta
M20MA005
Dept. of Mathematics
IIT Jodhpur
gupta.62@iitj.ac.in

2nd Deepti Gupta
M21CS005
Dept. of CSE
IIT Jodhpur
gupta.91@iitj.ac.in

3rd Nayanita Saha
M21CS010
Dept. of CSE
IIT Jodhpur
saha.5@iitj.ac.in

I. FIVE PHASES ARE IN GENETIC ALGORITHM

- Initial population
- Fitness function
- Selection
- Crossover
- Mutation

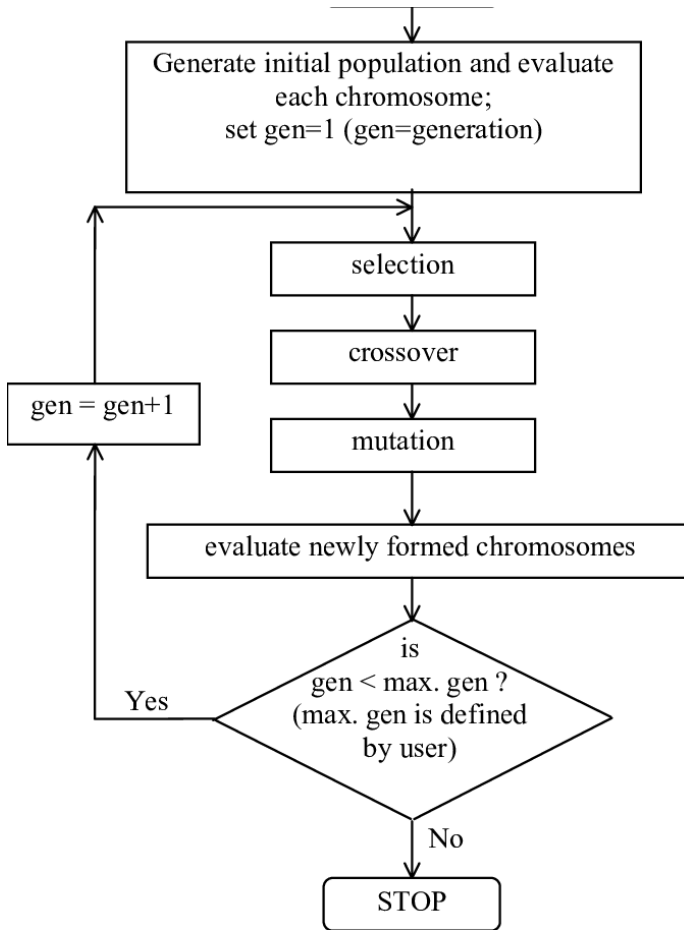


Fig. 1. Genetic Algorithm flowchart

II. INTRODUCTION

Neural Architecture Search is searching problem for searching the best architecture for the given network.

III. SEARCHING TECHNIQUE

For this project we have used **genetic algorithm**. A general flow of genetic algorithm has already been given. Following are the steps that have been used to solve the problem statement

1. Since we are supposed to optimise the parameters of the given CNN network, a general CNN network is made considering the constraints mentioned.
2. First the parameters are initialized with a range of available choices.
3. The Roulette fitness function is written by taking the weight(importance) of each species and selecting two parents with max fitness value.
4. Crossover has been done in different way. Instead of flipping parameter value one child with one parent, we are randomly deciding which the parent to be flipped.
5. Mutation/flipping has been achieved by changing the gene of children by changing the epoch(adding some integer to existing epoch number). This differentiates the two consecutive children.

Process

1. Instead of using the code snippet for downloading dataset, we have downloaded from keras and did standard train(60000) and test(10000) split.
2. The classification accuracy is calculated on test set.
3. The whole process take place in **3 generation having 4 step each**. The initialised parameters are passed to CNN network and classification accuracy is calculated which is followed finding parents ->crossover ->mutation.
4. This whole process repeats for three generations. Moreover in each generation, we have removed 2 species having lowest fitness value. This is done to ensure only fittest species is selected as parents.

The convergence criteria is selected when the accuracy is greater than 95% or when 3 generations are done. While running the code we found that 3 generations are sufficient for getting 75-85%.

Accuracy and Parameters trade off

For minimising parameters we have restricted our search space of filter size in the range of (16,32,64). We have not taken other higher filters as it will increase model complexity and thus parameters. **Also we have checked the condition, number of parameters less than 60000**(below 60000 searching time exceeds 2.5 hrs). Since our search algorithm should not stop at 75% so we took the cutoff of 95%. Randomness and change in crossover improved accuracy on test set and reducing search space and setting threshold minimised parameters.

IV. BEST ARCHITECTURE

The best architecture returned by our searching algorithm is:

1. In normal layers(NC) filter size = 16, kernel size = 6, activation = relu
2. In 2 reduction layers(RC) filter size = 64, kernel size = 3, activation = tanh
3. In dense layers activation = sigmoid

Final genome string we got: **NC 16 6 relu;RC 64 3 tanh;RC 64 3 tanh;FL sigmoid;**

Total number of parameters: **51610**

Test set accuracy: **81.84%**

V. REFERENCES

1. Introduction to Genetic Algorithms- towards data science
2. A Simple Genetic Algorithm from Scratch in Python- towards data science
3. Genetic algorithm basics
4. Genetic Algorithms Explained: A Python Implementation