Comparison Between Error Back Propagation and Genetic Algorithm for XOR Problem

Parnian Taheri[a]

[a] *Sharif University of Technology, , Tehran, , ,*

Abstract

In this homework, we classify the XOR problem using mlp classifier. In order to update the weights, we utilize different methods such as Error Backpropagation and Genetic algorithm. In the following, the methods are compared and the results are shown.

1. Introduction

XOR problem is one of the famous problems in classification because it cannot be classified using one line, therefore, different methods are proposed to solve this problem. In this assignment we use 2:2:1 mlp as our classifier. The aim is to compare different methods for updating the parameters and see if the model converges to the desired output. The techniques that we use are Error Backpropagation and Genetic algorithm. The reason for choosing these methods is that these are two popular and common methods that are used in different problems. In the following, a brief introduction for both methods is provided and the result of each is given to see which one works better.

2. classifier

2.1. MLP

A Multilayer Perceptron (MLP) is a type of artificial neural network that consists of multiple layers of interconnected nodes, or neurons. It is a feedforward neural network, meaning that the information flows through the network in one direction, from the input layer through hidden layers to the output layer.
The input layer receives the initial data or features. Each node in the input layer represents a feature of the input data, and the number of nodes in this layer corresponds to the dimensionality of the input space. We have two neurons in the input layer, which takes binary numbers.
The input and output layers are connected through hidden layer. These connections have associated weights that are adjusted during the training process. We have also two neurons in hidden layer with the sigmoid activation function and the sum of products as their input.
The output layer produces the final result or prediction. The number of nodes in the output layer depends on the nature of the task, such as binary classification (one node), multi-class classification (multiple nodes), or regression (one or more nodes). In our assignment, there is only one neuron as output.

3. Update Methods

3.1. Error Backpropagation

In this method, in order to update the weights, we first initialize the parameters and assign a random number between (-5, 5) to each weight and then starting from the outer layer, we update the parameters and move backward to reach the input layer. We continue this process until the desired output appears, or it runs for a given epoch. In this assignment we run the model for 1000 epochs, however the model stops when the error is less than 0.03. The error that we used is mean square error(MSE). We use the following formula to update the output and hidden weights respectively:

$$\Delta w^l_{u_{out}} = \eta(o^l_u - out^l_u)out^l_u(1 - out^l_u)in^l_u$$

$$\Delta w^l_{u_{hidden}} = \eta \sum_{s \in succ(u)} (\delta^l_s w_{su})out^l_u(1 - out^l_u)in^l_u$$

which η is the learning rate and o^l_u is the desired output.

3.2. Genetic

Genetic algorithms (GAs) are optimization and search algorithms inspired by the process of natural selection and genetics. They are part of a broader class of algorithms known as evolutionary algorithms. In this method, we first initialize variables in a way that a population of potential solutions (individuals or candidates) is randomly generated. Each solution is typically represented as a set of parameters, often referred to as a chromosome or genotype. Each individual in the population is evaluated using a fitness function. The fitness function quantifies how well an individual solves the problem at hand. It could be a measure of how close the solution is to the optimal solution. Individuals are selected to become parents based on their fitness. Moreover, mutation and crossover are used to introduce new population. In crossover, pairs of selected individuals (parents) exchange information to create new individuals (offspring) however, in mutation, random changes are introduced to the offspring's genetic information. finally, the new individuals (offspring) replace the least fit individuals in the population. This maintains the population size and ensures that better

solutions have a higher chance of persisting in subsequent generations. The algorithm iterates through these steps for a predefined number of generations or until a termination condition is met. Termination conditions may include reaching a satisfactory solution, a fixed number of generations, or a certain level of convergence.
In our model, the size of the chromosome is 9 equal to the number of weights. During each generation, one child is generated from parents, which crossover and mutation are applied on it randomly and the fitness is 1/MSE (Mean Square Error).

4. Results

In this section, the accuracy of each pair of the four classifiers and two feature extraction techniques on the Arabic digit recognition is presented. Moreover, the KNN classification technique have a parameter that needs to be adjusted, which is discussed below.

4.1. Accuracy

After running the model with these two methods for random seed = 42, the result shows that both methods can converge to the desired output, however, the GA method converges with less iteration and speed.

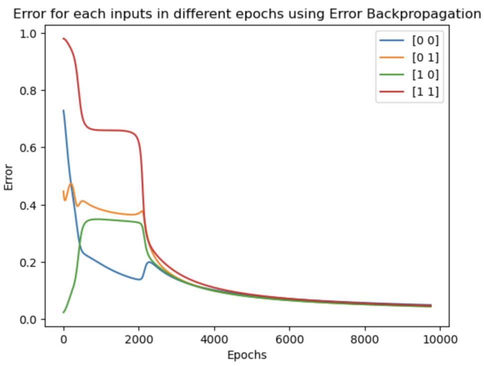

Figure 1: Error Backpropagation

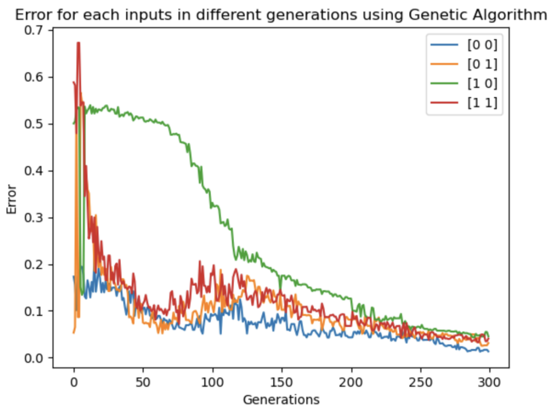

Figure 2: Genetic Algorithm