Grey Wolf Optimizer for Multi-Layered Perceptron

Sai Prathik Vasireddy
Computer Science Engineering
BML Munjal University
Gurgaon,Haryana
vasireddy.saiprathik.18cse@bmu.edu.in

Kireet Gannavarapu
Computer Science
BML Munjal University
Gurgaon, Haryana
g.kirannagakireetachary.18csc@bmu.edu.in

Venkata Abhiram Edapalapati
Computer Science Engineering
BML Munjal University
Gurgaon,Haryana
edapalapati.venkataabhiram.18csc@bmu.edu.in

***Abstract*— Bio-inspired algorithms are algorithms which are inspired from how nature solves a lot of its problems. A neural network needs proper weights on its neurons to efficiently solve a problem. This optimization of the weights can be a tedious task. In this paper we are discussing a Bio-inspired algorithm called Grey wolf algorithm for training a neural network and optimizing its weights. Our results show how the Grey wolf optimizer is accurate and how it does not get stagnant at local minimas.**

Keywords*— *Grey wolf optimizer, Multi-layer perceptron, Neural network, Regression

I. Introduction

In recent years, Neural networks have created a lot of excitement in the domain of computer science and many others. The basic concept of neural networks were proposed first in 1943, but due to the lack of hardware good enough to make neural networks shine, they were not noticed much. But now, due to advances in computer architecture, their applications spread to almost every domain available. A Neural network essentially takes a real world problem and tries to formulate a mathematical representation of the problem. By doing this, we can find the solution to any input given to the neural network

The core part of neural networks are the weights present on each of the neurons, proper values for these weights are important for creating a neural network which can accurately approximate the real world problem. Thus, finding the optimal weights on the neurons is an important step.

The algorithms which are used to optimize these weights are called trainers which train the model to give accurate results.

Bio-inspired optimization algorithms are the algorithms which are inspired by nature and are some of the algorithms which work really well for training a neural network. Some examples of these algorithms are Ant colony optimization, Particle swarm optimization, Cuckoo search optimization. One such algorithm is the Grey Wolf Algorithm. The reasoning behind using this algorithm is because of its high accuracy and performance in terms of approximating the global optima. Unlike other bio-inspired algorithms, GWO does not have a local minima stagnation problem.

II. Multi-Layered Perceptron

Neural networks architecture essentially consists of three types of neurons which are used to make the network. These are input neurons, hidden neurons and output neurons. Multi-Layered Perceptron are a kind of neural network which essentially contain three parallel layers of neurons. One input layer, one hidden layer and one output layer. The output of a MLP is calculated by the following steps.

$$s_j = \sum_{i=1}^{n} (W_{ij}.X_i) + \theta_j, \qquad j = 1,2,..h \tag{1}$$

Where W_{ij} is the weight between ith input and jth output neuron and theta is the bias of each hidden neuron. Next, the output of each of the hidden neuron is calculated

$$S_j = \text{sigmoid}(s_j) = \frac{1}{(1+\exp(-s_j))}, j = 1,2,..h \tag{2}$$

The final outputs are defined by the following formula:

$$y_k = \sum_{i=1}^{h} (W_{jk}.\ S_j) + \theta_k, k = 1,2,..m \tag{3}$$

Where W_{jk} is the weight between jth hidden neuron and the kth output neuron.

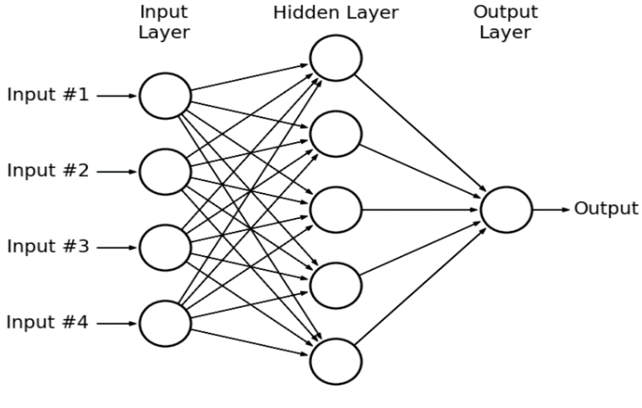

Fig. 1 Multi-Layer Perceptron

III. GREY WOLF OPTIMIZER

Grey wolf optimization (GWO) algorithm is one of the new meta-heuristic algorithms,it mimics the leadership hierarchy and hunting mechanism of grey wolves nature.

Fig. 2 Grey wolf

The entire population of Grey wolf is divided into four groups: alpha (α), beta (β), delta (δ) and omega (ω) which are ranked as their social hierarchy respectively. The optimisation process mainly consists of 3 levels i.e. searching, encircling and attacking. Alpha wolf acts as the leader of the group. Beta wolf and Delta wolf assist the alpha wolf in controlling the group movements. Omega wolves follow alpha, beta and delta. At the start of the hunting, alpha is considered to have the best possible location / solution.

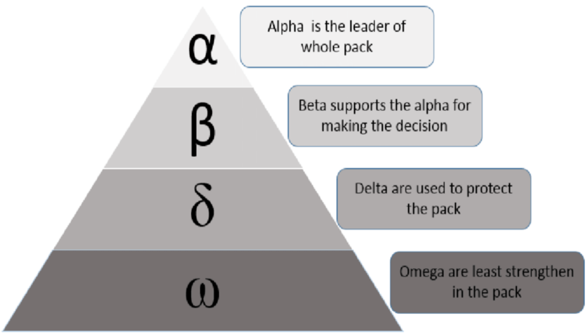

Fig. 3 Grey Wolf Hierarchy

A. Encircling

Grey wolves encircle the prey in the process of hunting and they reduce the distance between the prey and themselves gradually by following the leading wolves direction. The encircling behaviour of the wolves is mathematically represented as follows

$$\vec{D} = \left|\vec{C}.\overrightarrow{W_P}(t) - \vec{W}(t)\right| \quad (4$$

$$\vec{W}(t+1) = \overrightarrow{W_P}(t) - \vec{A}.\vec{D} \quad (5$$

Where D is the distance from omega and alpha/beta/ delta, W_P is the position vector of the prey W is the position vector of the wolf, t is the current iteration number and A, C are coefficient vectors. The values of A and C are found as follows

$$\vec{A} = 2.\vec{a}.\vec{r_1} - \vec{a} \quad (6)$$

$$\vec{C} = 2.\vec{r_2} \quad (7)$$

A is the random value in the interval [-2a, 2a], where components of a are linearly decreased from 2 to 0 over the course of iterations and r1, r2 are random vectors in [0, 1].

B. Hunting

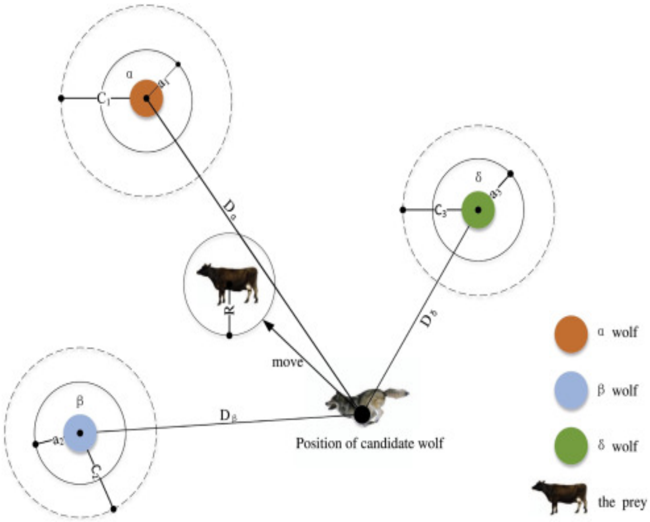

Fig. 4 Grey Wolf Pack Hunting a prey

The hunting in a Grey wolf pack is usually guided by the Alpha wolf. We can represent this phenomenon mathematically, the top three best solutions are saved and are used to update the Omega position according to Alpha, Beta, Delta with the help of the following equations:

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}\right| \quad (8)$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2}.\overrightarrow{X_\beta} - \vec{X}\right| \quad (9)$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}\right| \quad (10)$$

Where, D_α is the distance between alpha and omega, D_β is the distance between beta and omega and D_δ is the distance between delta and omega.

$$\overrightarrow{W_1} = \left|\overrightarrow{W_\alpha}. - \overrightarrow{A_1}.\overrightarrow{D_\alpha}\right| \quad (11)$$

$$\overrightarrow{W_2} = \left|\overrightarrow{W_\beta}. - \overrightarrow{A_2}.\overrightarrow{D_\beta}\right| \quad (12)$$

$$\overrightarrow{W_3} = \left|\overrightarrow{W_\delta}. - \overrightarrow{A_3}.\overrightarrow{D_\delta}\right| \quad (13)$$

$$\overrightarrow{W_{(t+1)}} = \frac{(\overrightarrow{W_1} + \overrightarrow{W_2} + \overrightarrow{W_3})}{3} \quad (14)$$

Where $W_{(t+1)}$ is the updated weights of omega which is the average of W_1, W_2, W_3

C. Attacking

Grey wolves finish the process of hunting by attacking the prey only when it stops moving. For mathematically modeling the approach of the prey we decrease the value of a from 2 to 0 over the course of iterations. When random values of A are in [-1, 1], the next position of a search agent can be in any position between its current position and the position of the prey.

IV. GWO In MLP3

An important step in training an MLP using meta-heuristics is the problem representation, i.e. the problem of training MLPs should be formulated in a way that is suitable for meta-heuristics. weights and biases are the most important variables in the training of an MLP. An optimizer should find a set of values for weights and biases that performs the best classification, approximation, prediction with good accuracy.

The GWO algorithm accepts the values in the form of a vector i.e. positions or co-ordinates, hence the weights and bias vectors are converted into matrix and fed into the optimizer.

$$\overrightarrow{W} = \{\vec{u}, \vec{v}\} = \{W_{11}W_{12}W_{13} \dots \dots W_{nn}\} \qquad (15)$$

As previously mentioned, the objective in training an MLP is to find weights and biases that can perform the best classification or prediction with good accuracy for both training and testing samples. A common metric for the evaluation of an MLP Performance is the Error. In this metric, a given set of training samples are fed into the MLP and the following equation calculates the difference between the desirable output and the value that is obtained from the MLP.

The Pseudocode for GWO is as follows

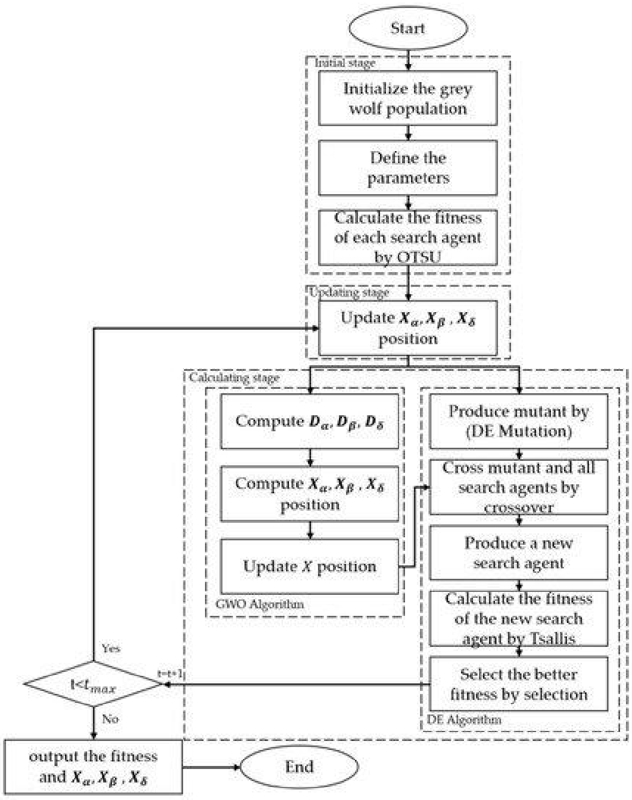

Fig. 5 Grey Wolf Optimiser Pseudo Code

V. Dataset

The Dataset that is used for training the MLP is a weather history dataset which contains the history of Temperature in Centigrade). Apparent Temperature is the focused variable which the model is meant to train and use to predict. Apart from that there are other variables like Pressure, Temperature, Wind Bearing, Wind speed and humidity.

VI. Results

We have tested our Grey Wolf Optimizer on the weather dataset and the following were observed.

Wolves Count	**10**	**20**	**50**	**10**
No.of Iterations	50	50	50	1000
Training Error	0.4107	0.2463	0.1937	0.1697
Validation Error	0.4137	0.2457	0.1950	0.1718

Tab. 1 Training and Validation errors for different test Cases

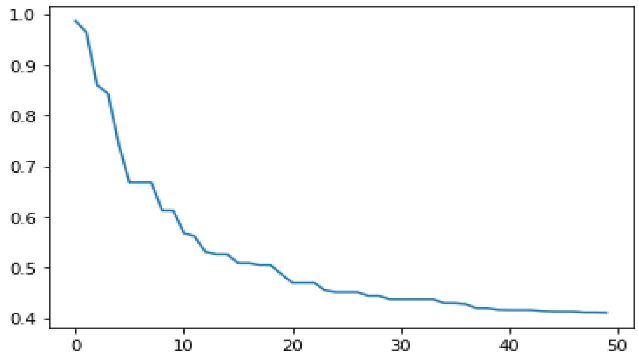

Fig. 6 Error graph for 10 agents

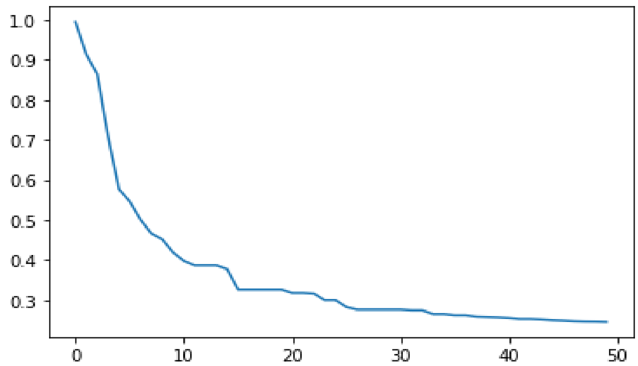

Fig. 7 Error graph for 20 agents

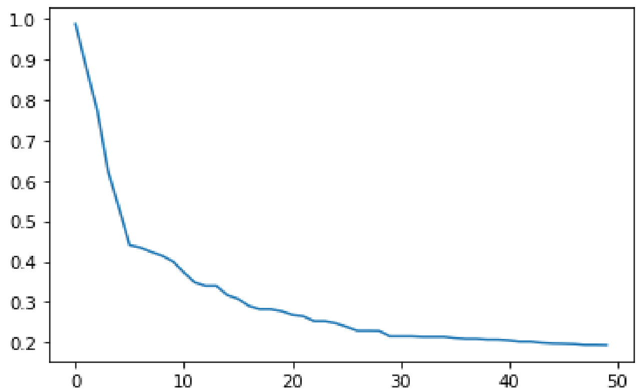

Fig. 8 Error graph for 50 agents

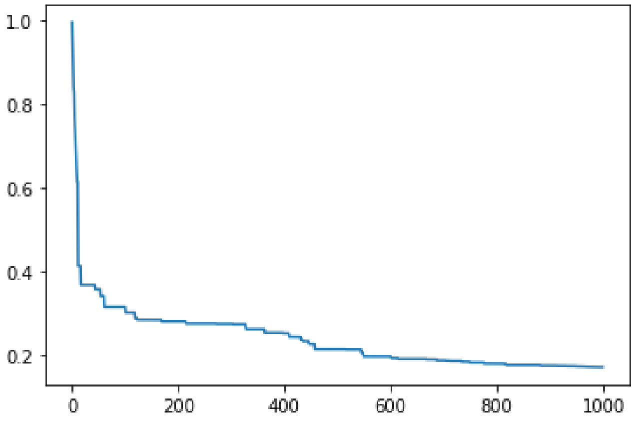

Fig. 9 Error graph for 10 agents and 1000 iterations

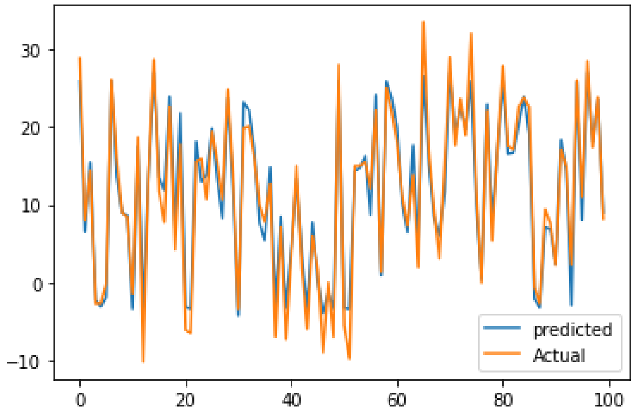

Fig. 10 Model output for 100 agents and 50 iterations

VII. Conclusion

In this paper, we have implemented the Grey wolf Optimization algorithm on a Multi-Layered perceptron to optimize the weight and bias parameters on our neural network in order to make a neural network to give the minimal error for the real world function it was approximating. From the results we can conclude that the Grey Wolf Optimizer can obtain a global minima with very low error and high accuracy. It also shows how the algorithm solves the problem of local minima and how it does not stagnate at a local minima.

It is observed that the algorithm can achieve higher accuracy on lower computation machines by decreasing the wolves count and increasing the iterations count.

References

[1] R. Rojas: Neural Networks, "Neural Networks" (A Systematic Introduction) Springer-Verlag, Berlin, 1996

[2] Bebis G, Georgiopolus M "Feed forward neural networks. Potentials",IEEE 13:27-31,1994

[3] S. Mirjalili, "How effective is the Grey Wolf optimizer in training multi-layer perceptrons", Applied Intelligence, In press, 2015

[4] Hossam Faris, Ibrahim Aljarah, Mohammed Azmi Al-Betar & Seyedali Mirjalili, "Grey wolf optimizer: a review of recent variants and applications"

[5] S Binitha, SS Sathya, "A survey of bio inspired optimization algorithms"

[6] Fevrier Valdez, "Bio-Inspired Optimization Methods"