International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Evolutionary Strategies for Parameter Optimization in Deep Learning Models

Dr. Shalini S.[1*], Aditya Kumar Gupta[2], Dr. Kiran Mayee Adavala[3], Ahmad Tasnim Siddiqui[4], Dr. Rohan Shinkre[5], Prasanna P. Deshpande[6], Manoj Pareek

Submitted: 16/07/2023 **Revised**: 09/09/2023 **Accepted**: 23/09/2023

Abstract: Evolutionary algorithms (EAs) have gained significant optimization techniques for deep learning model parameter tuning. Deep learning models often contain many parameters, and finding the optimal values for these parameters. EAs, inspired by natural evolution and natural selection processes, provide a promising approach for automatically searching and optimizing the parameter space. This study explores the application of EAs for deep learning model parameter tuning. We present an overview of deep learning and the challenges associated with parameter tuning. Next, we briefly introduce evolutionary algorithms and their key components, such as population initialization, reproduction, and selection operators. We discuss various strategies for integrating EAs into the parameter tuning process, including using different genetic operators, such as mutation and crossover, and techniques for handling constraints and incorporating prior knowledge. We aim of our present work is to optimize these hyperparameters using swarm optimization algorithms and evolutionary algorithms.

Keywords: *swarm optimization; evolutionary algorithms; hyperparameters; deep learning.*

1. Introduction:

Deep learning has emerged as a powerful technique for solving complex problems in various domains, including computer vision, natural language processing, and speech recognition. Deep neural networks (DNNs) excel at automatically learning hierarchical representations from large datasets, enabling them to make accurate predictions and extract meaningful patterns. However, the success of deep learning heavily depends on effectively tuning the numerous parameters associated with these models. Deep learning models typically consist of multiple layers with interconnected nodes, and each node has its set of weights and biases.

These parameters determine the behavior and performance of the model, making their optimization crucial for achieving optimal results.

However, the parameter optimization problem in deep learning is highly challenging due to several factors. Numerous ways are available to explore maximization and minimization difficulties; therefore, it is common for scholars to attempt to trace back a variety of modern difficulties to such two types. In addition to analytical approaches, there is a great deal of interest in solution space mapping approaches, including evolutionary computation and swarm optimization. In this study, we examine different method, including simplified swarm optimization (SSO), bacterial evolutionary algorithms (BEA), invasive weed optimization (IWO), particle swarm optimization (PSO), differential evolution (DE), and genetic algorithms (GA). As a result of our prior work with a number of these algorithms, adjusting and determining their parameters is not a difficult task. In recent times, gradient-based algorithms have been utilized in a variety of minimization-related domains, as well as in deep learning [7]. The vast neural networks employed in machine learning contain several parameters. Those parameters pertain to the network's architecture and its numerous algorithms. A crucial aspect of a correct solution is the proper selection of the parameters of deep learning algorithms, also known as hyperparameters. Modification of these hyper-parameters is often carried out by human professionals using mathematical reasoning and experimentation. The optimization technique is one of the most extensively employed optimization methods in this discipline [8]. Numerous ways available to explore maximization and minimization difficulties, therefore it is common for scholars to attempt to trace back a variety of modern difficulties to such two types. In addition to analytical, there is a great deal of interest in solution space mapping

*1*Associate Professor, DSATM, Bangalore, Shalini.siddamallappa@gmail.com*
2Associate Professor, School of Management Sciences Varanasi, UP aditya.guptas@gmail.com
3Faculty, Kakatiya Institute of Technology & Science kiranmayee@research.iiit.ac.in
4Associate Professor, (School of Computer Science & Engineering) Sandip University, Nashik, Maharashtra tasnim5@yahoo.com
5Research Consultant, Central Research wing, KLE Society's Institute of Dental Sciences, Bangalore rohanshinkre@gmail.com,
6Assistant Professor (Electronics and Communication Engineering) Shri Ramdeobaba College of Engineering and Management, Nagpur (India) deshpandepp@rknec.edu
7Associate Professor, Bennett University, Greater NOIDA, India manoj.pareek@bennett.edu.in

approaches including such evolutionary computation and swarm optimization. This research, we examine different method, including Simplified Swarm Optimization (SSO), Bacterial Evolutionary Algorithm (BEA), Particle Swarm Optimization (PSO), Invasive Weed Optimization (IWO), Differential Evolution (DE), and Genetic Algorithm (GA). As a result of our prior work with a number of the algorithms, adjusting and determining their parameters is not a difficult task.

2. Related Works:

This section presents the relevant studies, applicable approaches, and studied issues.

2.1 Evolutionary Algorithms:

Many lists of the most important are based on natural systems. The benefit of the techniques is their capacity to solve and different-optimize like discontinuous problems, multi-modal, high-dimensional, and nonlinear. Evolutionary algorithms have indeed been demonstrated to be effective at tackling multi-objective, non-linear, and constrained optimization. Such methods have had the capacity to investigate vast acceptable regions while requiring objective function deviations, as gradient-based training approaches. Their guiding ideas are founded on the hunt for a population of results, with fine-tuning accomplished through processes analogous to physiological reproduction. Humans are assessed and ordered using the fitness value in evolutionary computation. The genetic algorithm [1] is among the best-known evolutionary algorithms. When creating new organisms, the mutation and crossover operators are utilized. In contrast, the microbial proposed method [2] employs bacterial mutations and genetic transfer operators. Typically, evolution operators can either introduce new individuals to the community or modify current members. In various implementations, the essential characteristics of the algorithms are either one or the other. The operator in Differential Evolutionary [3] is based on the distinction between numerous randomly picked members of the population. The invading weed optimizer [4] simulates the behavior of weed colonization. Each seeds generate seeds proportional to its fitness, and the created seeds are distributed at random across the search area.

2.2 Swarm Optimization:

In swarm optimization approaches, no new recruits are ever added to the populace; only member nations are modified. Utilizing their own experience and the experience of the community as guides, participants explore their surroundings in an effort to discover ever-better locations. Optimization of particle swarms is among the most well-known swarm optimization techniques [5]. The particles inside the search area are in motion. They remember both their own best point and the best point of the entire swarm (the social component) within the area of search and have a velocity vector. The new velocity vector is created using those three elements. In simplified optimization technique [6], the new role vector is created instantly based on the various components.

2.3 Multilayer Perceptron:

A MLP is type of feedforward neural network. A Multilayer Perceptron has at least three layers of node including an output layer, a hidden layer, and an input nodes. Apart from the input neurons, everything node in the graph a neuron whose activation function is irregular. For train, MLP employs backpropagation, a supervised training approach. MLP differs from Activation functions by virtue of its many layers and activation functions function. It is capable of learning non-linearly separated data. [9]

2.4 VGGNet:

VGGNet is created by the Visual Geometry Group (VGG) at Oxford University [11]. Though VGGNet is the marathoner in the categorization challenge of the 2014 ILSVRC (ImageNet Large Scale Visual Recognition Competition), it was not the champion. This net is highly structured and distinct, thus you chose it for optimization.

2.5 MNIST, Fashion-MNIST:

Systems for image processing are typically trained using the MNIST database, a massive collection of numbers that are handwritten [10]. The NIST black and white images were generally pro, thereby added gray levels, then adjusted to fit within a 28x28 pixel bounding box. 10,000 testing data and 60,000 training images are current in the Pertained . The e-commerce startup Zalando generated the Fashion-MNIST dataset, which substitutes fashionable photos for handwritten characters [12]. The size of the image and division structure between testing and training is the same. Figure 1 shows the Fashion-MNIST dataset in detail.

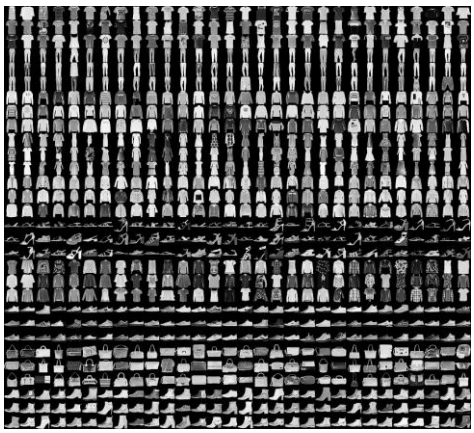

Fig.1 MNIST fashion dataset

3. Optimization of Hyperpara Meters:

We used Python to create a framework and execute swarm algorithms and the evolutionary. The ability to readily parallelize optimization techniques over Bayesian optimization is one of its benefits. For simultaneous training of the network throughout the search procedures, we employed this functionality and four Nvidia GeForce GTX 1080 Ti graphics cards. While the value of the hyperparameters are frequently integers, the so-called genes in most Darwinian and swarming optimization techniques are actual figures. So, when developing individuals personal genotype, rounding down is done before considering a possible solution.

3.1 MLP on MNIST:

An MLP must initially be optimized, and the search algorithm can do this by modifying the following hyperparameters:

- Rate of acquisition: 0.0001 – 1.
- The number of concealed layers ranges from 0 to 4.
- Dropout: 0 – 0.9.
- Size of hidden units: 5 to 1495.

The following are the parameters of the employed algorithms:

- DE: CR = 0.6, F = 0.8.
- WO: e = 2, itermax = 100, Nmax = 6, σinit = 0.18, Nmin = 1, σf in = 0.05.
- PSO: φg = 4, φp = 2, ω = 1.
- SSO: Cg = 0.8, Cp = 0.4,Cw = 0.2.

The IWO algorithm's populace has eight chromosomal (individuals), whereas the other methods have twelve. Adam optimizer [13] is used to train the neural network. Each layer is preceded by a dropout layer; hence, one chromosome includes 11 genes. Not always were chromosomal components utilized. Inside the case of the training data, the base 10 power was altered from 0 to 4. Even as loss function, cross-entropy is applied. Your training data were normalized within the range of 0 to 1. No additional preparation was done to the information. The retraining was concluded whenever the train loss did not decrease during the course of 14 epochs. When there were no improvements after five epochs, then training rate was reduced by one-fifth. The termination condition of the simulated annealing is when the population's esteem and prestige acquired doesn't really improve after 10 iterations. The fitness function penalizes neural networks (NN) with more learnable parameters. For, the fitness function defined by Equation (1) has two components: the accuracy component and the parameters component.

Parameters = log10 (number of parameters),

Accuracy = 100 − (validation accuracy ∗ 100),

Fitness value = accuracy part + (parameters part/5) (1)

The functional can equilibrate any two components. It permits the development of about one layer with a 1% improvement. Four algorithms were utilized to solve this problem like DE, SSO, IWO and PSO. In Table I lists the parameters that have been optimized. The results of the optimization are displayed in Table 2 like fitness value (FV), number of evaluations (NOE), number of parameters (NOP), and validation accuracy (ACC)

Table 1: Optimized MLP parameters

	LR	DO	SOL	NHL
SSO	7.8e-5	0.24, 0.39, 0.08	1356, 956	3
PSO	2.2e-4	0.32, 0.08	700	2
IWO	3.2e-5	0.42, 0.04	1323	2
DE	4.2e-4	0.34, 0.03	623	2

Table.2: MLP optimization outcomes:

	FV	NOE	NOP	ACC
SSO	2.24	270	3 184 506	99.98%
PSO	2.22	265	23187	98.23%
IWO	2.21	245	187 495	98.45%
DE	2.20	240	440 440	98.67%

These test dataset accuracy rates are comparable to the outcomes of employing similar network without processing the material data [10]. Utilizing the suggested fitness function, network was discovered SSO, whose precision is little lower the system discovered DE, but its size is five times larger. SSO received the most fitness-related calls. The Figure 1 illustrates the architecture of the optimal MLP.

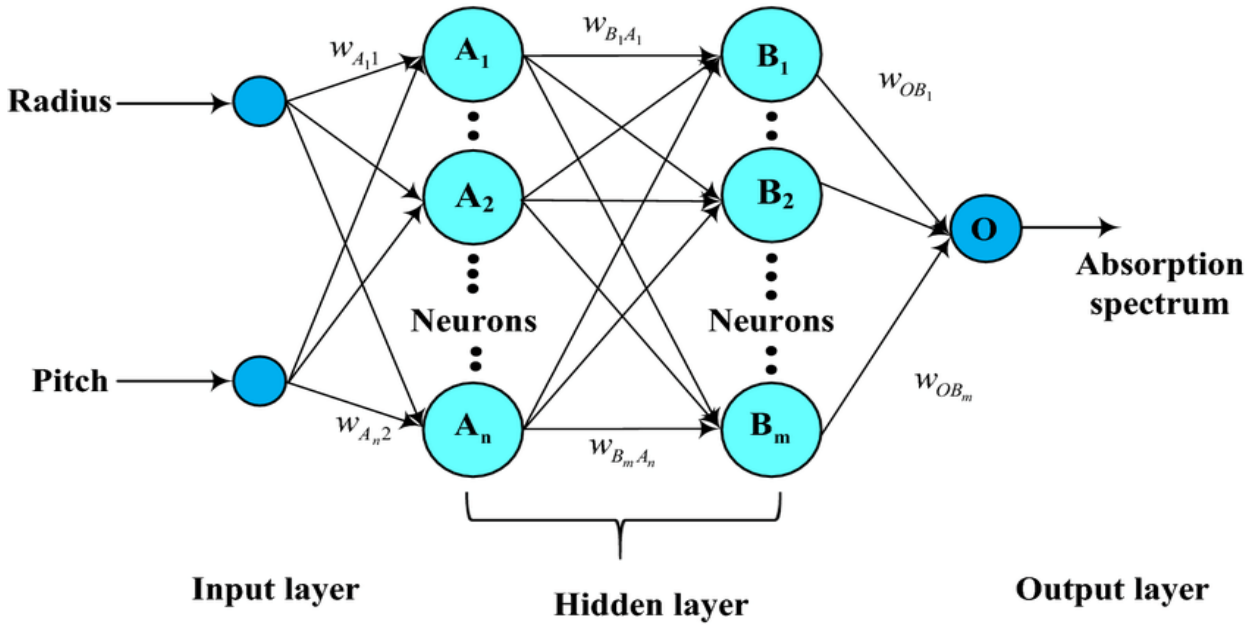

Fig.1 an MLP structure that is optimum

3.1 VGG about Fashion-MNIST:

Today we examine the paper's primary objective, which is the optimization of the architecture of VGG16-like network on the Fashion MNIST dataset. The termination circumstances are identical to the past segment with a few modifications. The researcher find evolution and swarms techniques best global, to improve for 5 iterations, and the patient throughout the NN training was originally 8 epochs following by a 12 epochs focused search.

The search algorithm is capable of modifying the following hyperparameters:

- Optimizer (O): SGD, Adam, RMSProp,
- Activation function (A): Tanh,, ReLU,
- Dropout (DO): 0 – 1,
- Number of filters (F): 1 – 51,
- Number of filters (F): 1 – 51,
- Number of convolution blocks (NB): 1 – 3,
- Size of fully connected layer (SD): 1 – 501,
- Number of convolution layers in one block (NC): 1 – 6,
- Convolution kernel size (KS): 1 – 10,
- Fully connected layers (ND): 1 – 7,

Literature reveals that the best Fashion-MNIST test results are between 96% and 97%, but preprocessing was predominantly used [12]. This dataset has a human (non-expert) accuracy of 84.5%. More specifically, the literature results for a VGG16 26 M networks are 94.5%. The only modification we made to the data was to normalize it between 0 and 1 as show in figure.2.

The applied evolutionary and swarm algorithm parameters:

- IWO: e = 2,itermax = 10, Nmin = 1, σinit = 32, σf in = 8,
- GA: pmut = 0.7,
- SSO: Cg = 0.9Cw = 0.25, Cp = 0.5,
- BEA: Nclones = 2,
- PSO: φg = 3, ω = 1.5, φp = 2,
- DE: CR = 0.3

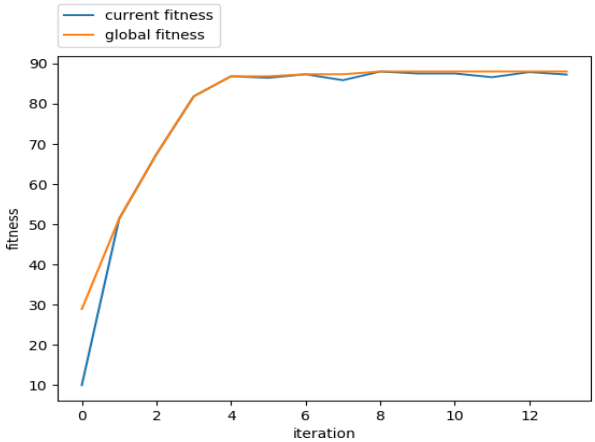

Fig .2 : SSO Fitness Value

In this job, we did not penalize larger neural network models. Therefore, channel of the fitness function computed accuracy .In the case of algorithms with a smaller projected quantity of fitness callbacks, we raised the number of people to achieve better outcomes. Table 3 presents the results obtained like number of iterations (NOI), number of chromosomes (NOC), validation accuracy (ACC) and number of assessments (NOE).

Table 3. Optimization results of VGG

	NOE	NOC	NOI	ACC
SSO	200	5	23	90.98%
PSO	234	6	21	89.23%
IWO	243	7	11	88.45%
DE	221	8	10	87.67%

Above 90% accuracy is already an appropriate performance for GA and IWO. In light of these outcomes of fitness function invocations, the swarm-based approaches and GA appear promise for achieving a precise result in a fair amount of time. Figure 3 depicts, by way of illustration, a simulator utilizing the SSO approach. In the graph, greater fitness corresponds to a value that is closer to 100. Orange represents the world's highest level of fitness. This refers to the ideal solution discovered throughout the optimization process. In contrast, the color blue reflects the best result for the existing population. In elitism algorithms, these two functions are identical.

In Tables 4 and 5, the acquired parameters from the best runs are displayed

Table .4 Optimized VGG parameters (I)

	NC	NB	LR	DO	WD
IWO	5	2	0.087	0.083	0.082
GA	4	2	0.234	0.234	0.23
SSO	3	4	0.456	0.455	0.45

Table .5: Optimized VGG parameters

	ND	SD	KS	F	A	0
IWO	5	186	8	45	ReLU	Adam
GA	4	189	3	43	ReLU	RMSProp
SSO	3	177	11	42	ReLU	NAdam

The table statistics have a broad range, allowing for a variety of successful parameter selections. By conducting additional simulations, five algorithms are studied further. The maximum number of seedlings (individuals) is raised to seven in the case of IWO dropped to 1, g is increased to 6, and the perform many different factor remains at 5. The GA, DE and SSO parameters were not altered. Table 6 presents the obtained results, while Tables 7 and 8 display the associated parameters.

Table .6 Refined VGG optimization results:

	NOE	NOC	NOI	ACC	NOP
SSO	200	5	23	90.98%	123456
PSO	234	6	21	89.23%	120234
IWO	243	7	11	88.45%	121345
GE	221	8	10	87.67%	45676
GA	233	13	14	98.34%	678907

Table .7 Enhanced VGG optimization parameters:

	NC	NB	LR	DO	WD
IWO	5	2	0.087	0.083	0.082
GA	4	2	0.234	0.234	0.23
SSO	3	4	0.456	0.455	0.45
GE	5	3	0.456	0.455	0.45
PSO	4	2	0.456	0.455	0.45

PSO could be improved, but the end outcome is still unacceptable. The outcome of Genetic Algorithm is acquired during the initial generation, and it took place no advancement over five generations of development that followed, therefore the search was discontinued. The outcome is good, it is accompanying network is very complex and contains an excessive number of trainable parameters. Surprisingly few factors produce a result that is identical to the initial structure. Its precision approaches the VGGNet's test precision. The Simplified Swarm Optimization is accuracy 93.57% is greater than the VGGNet of test accuracy; it employs slightly more parameters than IWO and DE, but its fitness function calls are over 80 fewer. In addition, SSO and DE selected a different course than IWO. It is intriguing that all solutions except DE used the "weakest" SGD optimizer. Figure 4 depicts the architecture of the most optimal VGG.

Table .8 Enhanced VGG optimization settings (2):

	ND	SD	KS	F	A	0
IWO	5	186	8	45	RELU	Adam
GA	4	189	3	43	RELU	RMSProp
SSO	3	177	11	42	RELU	NAdam
PSO	2	177	5	48	RELU	SGD
GE	3	177	8	35	RELU	SGD

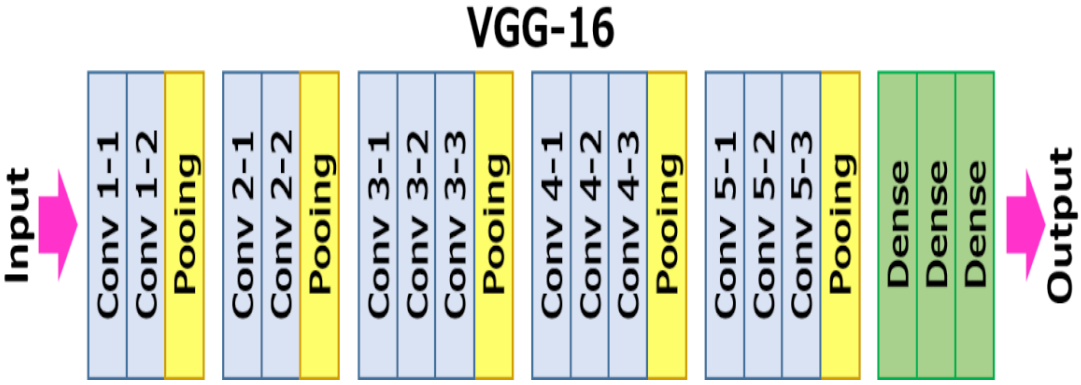

Fig.3 Structure of enhanced VGG

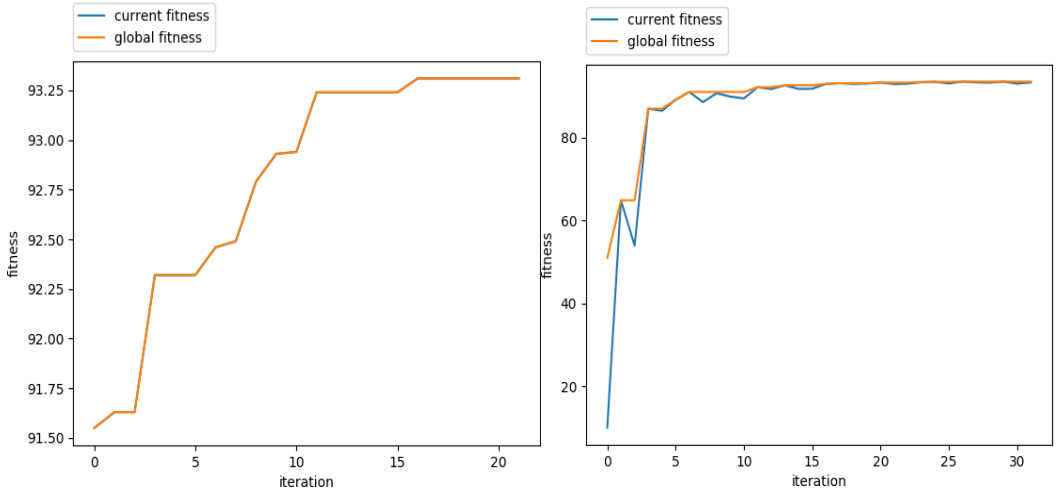

Fig .4: Depict the simulation results of the refined optimizations.

4. Comparing with Bayesian Optimization (Bo):

Bayesian Optimization using Gaussian Processes was also done on the two prior problems. This method evaluates initialization points before determining the next point to be assessed by a Gaussian process in the search space. In both instances, the algorithm was programmed to generate 16 starting points, and it terminates when the best fitness evaluation has not improved after 16 trials.The results of the best optimization runs are displayed in Tables 9, and 10.

Table 9: Results of Bayesian optimization

	ACC	NOE	NOP
MLP	98.79%	62	804 550
VGG	93.03%	40	1 030 923

The results resemble those of evolutionary algorithms. This method required more evaluations than evolutionary algorithms, but because it cannot be repeated several times even with more GPUs, its execution time was comparable to or slightly longer in our situation in show figure.5.

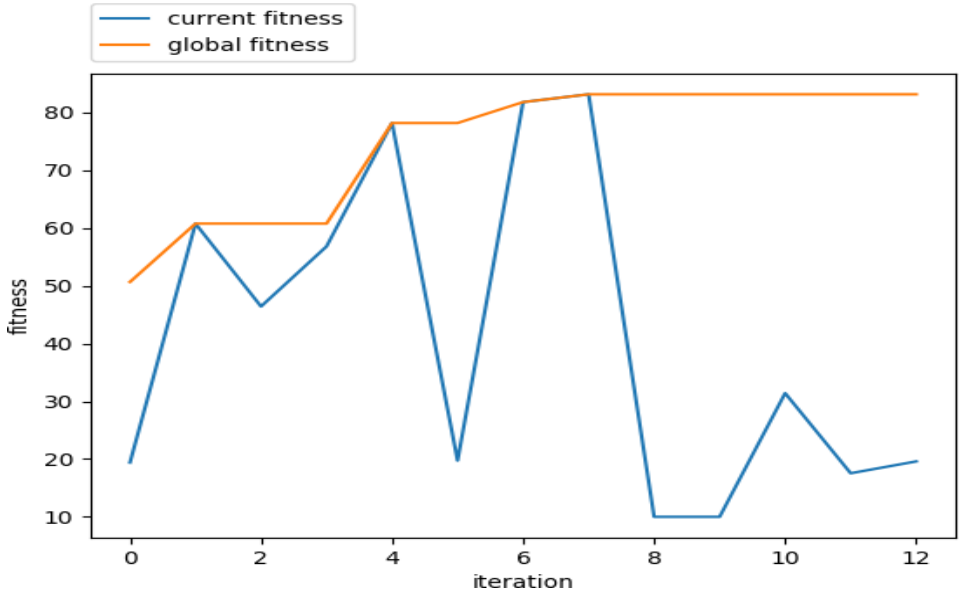

Fig .5: Value of health by PSO

Table.10 Bayesian optimized MLP parameters

NHL	SOL	DO	LR
1	1012	0.259	0.005

5. Conclusion and Future Work:

Parameter optimization is a critical aspect of deep learning, as the performance and generalization capabilities of deep learning models heavily depend on finding optimal values for the numerous parameters they possess. VGGNet can store up to 144 million parameters, depending on its configuration. The network generated by the SSO method is more optimum. The SSO approach for hyperparameter optimization based on the outcome, it obtained the good outcome and necessitates the same number of assessments per iteration based on size of population. The same evaluations was required for BO. Simplified Swarm Optimization is to optimally by making size of population based on GPUs; hence, it can evaluate more solutions simultaneously, thereby boosting the likelihood of locating superior answers. Each algorithm was only ran once; hence, the obtained results may be further enhanced in the future. To investigate their advantages and disadvantages, it will be necessary to execute the algorithms multiple times and to test them on increasingly more complex tasks.

References:

[1] J. H. Holland, Adaption in Natural and Artificial Systems. Cambridge, Massachusetts: The MIT Press, 1992.

[2] N. E. Nawa and T. Furuhashi, “Fuzzy system parameters discovery by bacterial evolutionary algorithm,” IEEE Transactions on Fuzzy Systems, vol. 7, no. 5, pp. 608–616, Oct. 1999.

[3] R. Storn and K. Price, “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[4] Mehrabian and C. Lucas, “A novel numerical optimization algorithm inspired from weed colonization,” Ecological Informatics, vol. 1, no. 4, pp. 355–366, 2006.

[5] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948.

[6] Bae, W.-C. Yeh, N. Wahid, Y. Chung, and Y. Liu, “A new simplified swarm optimization (SSO) using exchange local search scheme,” Int. J. Innovative Computing, Information and Control, vol. 8, no. 6, pp. 4391–4406, 2012.

[7] Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.

[8] P. I. Frazier, “A tutorial on Bayesian optimization,” www.arxiv.org/pdf/1807.02811.pdf, 2018.

[9] R. Hecht-Nielsen, Neurocomputing. Addison-Wesley, 1990.

[10] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[11] K. Simonyan and A. Zisserman, “VGGNet,” www.arxiv.org/pdf/1409.1556.pdf, 2015.

[12] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST,” www.arxiv.org/pdf/1708.07747.pdf, 2017.

[13] P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” www.arxiv.org/pdf/1412.6980.pdf, 2014.

[14] Supratak, H. Dong, C. Wu, and Y. Guo, “Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 25, no. 11, pp. 1998–2008, 2017.

[15] Sherje, D. N. . (2021). Content Based Image Retrieval Based on Feature Extraction and Classification Using Deep Learning Techniques. Research Journal of Computer Systems and Engineering, 2(1), 16:22. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/14

[16] Ch.Sarada, C., Lakshmi, K. V. ., & Padmavathamma, M. . (2023). MLO Mammogram Pectoral Masking with Ensemble of MSER and Slope Edge Detection and Extensive Pre-Processing. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3), 135–144. https://doi.org/10.17762/ijritcc.v11i3.6330

[17] Keerthi, R.S., Dhabliya, D., Elangovan, P., Borodin, K., Parmar, J., Patel, S. K. Tunable high-gain and multiband microstrip antenna based on liquid/copper split-ring resonator superstrates for C/X band communication (2021) Physica B: Condensed Matter, 618, art. no. 413203, .