THE PROPOSED ALGORITHM

A new PSO algorithm based on encoding weights by IP address is proposed.

A. Algorithm overview

The proposed algorithm is described in Algorithm 1.

Algorithm 1 Framework of proposed algorithms

$P \leftarrow$ Create the population using the IP encoding scheme
while Criterion did not meet do
    evaluate particles by the fitness function
    update each particle with detail in Algorithm 2
    compare with P^{best} and G^{best} and update if necessary
end while

B. The encoding strategy

Internet Protocol (IP) address is a combination of number for a device connected over the network [2]. IP address will have subnet networks, which often describe in Classless InterDomain Routing (CIDR) style [3]. IP address usually consists of 4 components, each being an integer. For example, a standard IPv4 address can be 146.25.168.3.

Although IP address is in the form of integers, separated by full stops, the underlying encoded bits are responsible for finding the right network. This characteristic inspires a new encoding scheme for neural network weights in PSO. To encode weights in the neural network, a decent number of bits is often required. As a result, the bits will create a large integer when transformed, thus limiting the convergence of PSO. This is because, for an encoded large number, only the latter few bits can be updated. The first few bits, which represent the larger value, will hardly get any update since it will require a big change in the value of the number. However, a continuously large change in value will make the convergence of PSO unstable. On the other hand, a small update will only update a portion of the bits, hence making PSO stuck in one place. Using the IP address encoding scheme, one huge integer can be separated into multiple bytes, each with a value from 1 to 256. One byte of the IP address will represent one dimension of the integers. This will increase the convergence rate of PSO, as all bits will have the chance to be updated
[4].

The algorithms are inspired by how IP address works. First, the number of bits represents each weight needed to be chosen. Then, each binary string will be separated into 2 bytes in the IP address. Each byte will cover one dimension of the particle. After encoding each weight into 2 bytes, the particle and velocity of PSO can be created.

For example, the encoding process is shown in Table I. The weight will be encoded into 16 bits, then separated into 2 parts, each consist of 8 bits. Then it will be decoded into integer, as one part of the IP address.

TABLE I
WEIGHT ENCODING

Weight	Weight after encoded	Bi-partition	IP address
0.06537993	1100110 001110001	11001100. 01110001	136.94

These 2 numbers will be 2 dimensions of a particle. As a result, 650 weights will turn into a particle with 1300 dimensions.

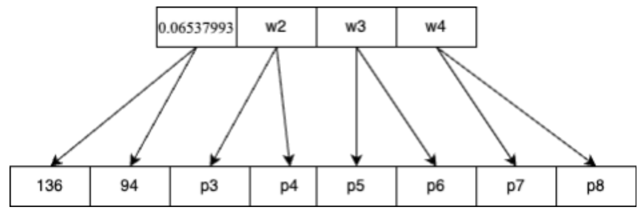

Fig. 6. The process of encoding weights into particle

C. Population

Each particle is created with dimensions of 1300. Each dimension is from 1 to 256. All the value of particle will be created randomly.

A population growth function is also introduced. In the early stages of particle swarm optimization, the PSO algorithm converges very quickly. However, as the evolution progresses, the algorithm's convergence speed slows down and the accuracy of the solution cannot be quickly increased. In addition, throughout the process, the size of the population remained unchanged. Another problem is the search time for the algorithms is slow. The introduction of a growth function will help the algorithm to obtain optimal solutions with less cost than traditional PSO [5]. After some iterations, a new particle is introduced to the population. This article will have the experience of the better ones, and ignore the characteristics of the worse ones. To do that, the new particle will have each dimension equal to the mean value of the 5 best particles' dimensions. For the growth formula, [5] suggests that logistic model will have the best performance:

$$x(t) = \frac{x_{\max}}{1 + \left(\frac{x_{\max}}{x_0} - 1\right) e^{-rt}} \tag{6}$$

where x is the population size at time t, x_{max} is the maximum population size, x_0 is the starting population size and r is the population growth rate.

D. Fitness Evaluation

Before evaluating the fitness, the particle is decoded. Two dimensions are combined into a string of 16 bits and then decoded into weights using real-value decoding. The range of weight is set as from -1 to 1. Then, the neural network will run predictions and we evaluate them based on accuracy.

A fitness evaluation scheme that only uses part of the data is set. For large datasets, the learning process is slow. In addition, as most of the computational cost is in the fitness evaluation, reducing that time will significantly improve the efficiency of PSO. Therefore, a new fitness evaluation method that uses only part of the dataset is proposed to dramatically reduce the learning time of PSO. From the original data, in each iteration, a random subset is chosen. The subset will contain 1000 pictures in each class. As a result, the subset will equal 1/5th of the original training data while having equal amount of each class, to make sure that the evaluation is fast and comprehensive.

E. Update Particle

Each particle is updated following Algorithm 2.

It is needed that the value of each dimension in the particle is in integer format. Therefore, a minor adjustment is introduced to set every after updating to an integer. Another adjustment is needed due to the restriction of the new encoding scheme. For the encoding to work, the particles need to remain inside the search space, which means that each dimension is from 1 to 256. If the particle jumps out of that for one dimension, the algorithms will malfunction as the encoding scheme will not work. In addition, there is a high chance that particle will move out of the boundary in the first few iterations if the search space is high-dimensional [1]. As a result, it will heavily affect the performance of PSO. Therefore, it is important to have a suitable mechanic to handle that situation. There are three possible handling schemes: Random, Absorb and Reflect. For random and absorbing schemes, the PSO algorithms may fail when solving high-dimensional and complex problems [1]. The random scheme will go against the idea of PSO, as the particle should move to the global best, not a random position. And the absorb scheme can limit the search space of PSO. If some of the dimensions of global and local best hit the boundary and get absorbed, the whole population will tend to move to that. As a result, some of the dimensions of other particles will also hit the boundary and get absorbed. This will limit the search space of PSO since the algorithms now can only search in a sub-space of the original search space. Therefore, the proposed algorithms will use the reflect mechanic. When a particle flies beyond a parameter's boundary, the boundary reflects the projection of the particles and functions as a mirror.

We update the particle's position and velocity by going through each byte and using the appropriate parameters. The new particle is evaluated after all the dimensions have been updated. Then, the fitness value will be compared with P^{best} and G^{best} to update them if necessary.

Algorithm 2 Update Particle

Input: particle individual vector *ind*, constant $c1$ and $c2$, w, v_{max} and v_{min}

Output: updated individual vector *ind*

for element interface in ind do
 x_i ← the i byte of the IP address in the interface;
 (r1,r2) ← uniformly generate $r1$,$r2$ between 0 and 1;
 $v_i(t+1)$ ← Update velocity based on Equation 1;
 $v_i(t+1)$ ← Apply velocity clamping using v_{max};
 $x_i(t+1) \leftarrow x_i + v_i(t+1)$
 if $x_i(t+1) > 256$ then
  $x_i(t+1) \leftarrow 256 - (x_i(t+1) - 256)$;
 end if
 if $x_i(t+1) < 1$ then
  $x_i(t+1) \leftarrow 1 - x_i(t+1)$;
 end if
end for
fitness ← evaluate the updated individual *ind*;
(P_{best},G_{best})← Update P_{best} and G_{best} by comparing their fitness;
return ind

F. Best Individual Selection

The best individual will be reported as the PSO's global best. That individual will contain a list of IP address, which are stored in the dimensions of the best individual. Every two bytes from left to right, can be decoded from the particle vector to get one weight of the final layer.

References

[1] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, Y. Liu, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, and Q. V. Le, "Symbolic discovery of optimization algorithms," 2023. [Online]. Available: https://arxiv.org/abs/2302.06675

convolutional-neural-networks-for-image-classification-f0754f7b94aa, 2023, accessed: 2024-12-04.

[2] J. Ancheta, M. Cortez, R. J. Diaz, D. Francisco, W. Torreno, and J. Villamor, "An overview of ip addressing," *Researchgate Publication 361053333*, 06 2022.

[3] V. Fuller, T. Li, K. Varadhan, and J. Yu, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy," RFC 1519, Sep. 1993. [Online]. Available: https://www.rfc-editor.org/info/rfc1519

[4] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification," *CoRR*, vol. abs/1803.06492, 2018. [Online]. Available: http://arxiv.org/abs/1803.06492

[5] X.-F. Liu, Z.-H. Zhan, and J. Zhang, "Incremental particle swarm optimization for large-scale dynamic optimization with changing variable interactions," *Applied Soft Computing*, vol. 141, p. 110320, 2023. [Online]. Available: https://www.sciencedirect.com/ science/article/pii/S1568494623003381