Manta Ray Foraging Optimization for Hyper-Parameter Selection in Convolutional Neural Network

Omar Adil Kamil1, *, Shaymaa W. Al-Shammari2

1 Department of Computer Engineering, Al-Nahrain University, Iraq
E-mail: omar.altae11@gmail.com
2 Department of Computer Engineering, Al-Nahrain University, Iraq
E-mail: shaymaa.alshammari@gmail.com

* Corresponding author: Department of Computer Engineering, Al-Nahrain University, Iraq. Tel: +9647702582489.

Abstract. Convolutional neural networks (CNNs) have been attracting attention as one of the most common deep learning techniques used for different applications like images classifications, objects recognition, face recognition, etc. The performance and efficiency of the CNN model depend directly on their hyper-parameter, which must be selected by an expert or using one of the models that proposed and improved. This makes it very important to determine the optimal hyper-parameters. In this work, the Manta Ray Foraging Optimization (MRFO) algorithm is used to select CNN’s Hyper-Parameter. We demonstrate that MRFO efficiently explores the solution space. This allowing CNN with simple architecture to achieve a good classification accuracy over Cifar_10 dataset. In the presented experiment Cifar_10 dataset used as the benchmark data sets. By optimizing CNN hyper-parameters with MRFO algorithm and comparing the obtained results with other CNN, it was approved that the accuracy was improved.

Keywords: Convolutional neural network; manta ray foraging optimization; hyper-parameters selection;

1. Introduction

Convolutional neural networks represent a special type of neural network. CNN is most commonly used in different visual tasks, such as classifying images, generating image descriptions, segmenting images, detecting objects in pictures, recognizing faces. CNNs’ models essentially consist of different layers, which are the convolution layer, the pooling layer, and the fully-connected layer (FC layer). In a deep network stacking these layers properly can successfully solve different visual tasks (Albawi, S., et al. 2017). The performance of any CNN models highly depends on the hyper-parameters such as filters, the number of convolutional layers, and batch size, etc. and require experts to determine these hyper-parameters, meaning that it might be hard to find good settings for a non-expert. In this paper, we propose solving the problem of selecting the hyper-parameters by using Manta Ray Foraging...
Optimization (MRFO) algorithm to obtain optimal hyper-parameters. The general hyperparameters in CNN models could be selected are the following: the number of filters, the number of convolutional layers, the stride, the size of the filters, the number of training epochs, activation function, the dropout probability, and the batch size.

This paper is organized as follows: Section 2 presents the related work, section 3 discusses the Manta Ray Foraging Optimization Algorithm, section 4 introduces the proposed method, section 5 shows the experiments and results and discusses the obtained results.

2. Related work

Recently, there have been many researches proposed a method to optimize the convolutional neural network hyper-parameter such as the research presented by (Serizawa, T. and H. Fujita 2020) who proposed a method for hyperparameter optimization of CNN by using linearly decreasing weights PSO (LDWPSO), which is metaheuristic algorithms and the model achieved better results than CNN without LDWPSO. Another hybrid method was presented by (Albeahdili, H. M., et al. 2015) called (PSO-SGD) algorithm training CNN. In (Lorenzo, P. R., et al. 2017) proposed a method using PSO to optimize the hyper-parameter in DNN. It shows that very simple DNN after optimizing by PSO can improve the performance of the model. Another interesting method was presented in (Junior, F. E. F. and G. G. Yen 2019) the authors proposed a method to search for convolutional neural networks (CNNs) architectures using particle swarm optimization. One more novel proposed by (R Takahashi, T Matsubara 2019) called a weight shared multi-stage network to improve classification accuracy. (Ismail, A., et al. 2019) presented a study for image classification using mini VGG Net architecture. The classification accuracy of the models increased by adding batch normalization layer to the network with learning rate decay schedulers to avoid overfitting. (Smith, L. N. (2017)) proposed a new method called Cyclical Learning Rates (CLR) to find best value for learning rate instead of using fixed value.

3. Manta Ray Foraging Optimization Algorithm

MRFO algorithm is a new optimization technique proposed by (Zhao, W., et al. 2020). The revelation of the algorithm is based on behaviors of manta rays. These creatures have three intelligent and fantastic foraging strategies.

1. Chain foraging uses this strategy by forming an orderly line and individually update the best position found can be expressed mathematically as follows:

\[ x_i^d(t + 1) = x_i^d(t) + r \cdot (x_i^b(t) - x_i^d(t)) + \alpha \cdot (x_i^b(t) - x_i^d(t)) \quad i = 1 \] (1)

\[ x_i^d(t + 1) = x_i^d(t) + r \cdot (x_{i-1}^d(t) - x_i^d(t)) + \alpha \cdot (x_i^b(t) - x_i^d(t)) \quad i = 2, ..., N \] (2)

\[ \alpha = 2 \cdot r \sqrt{|log(r)|} \] (3)

2. Cyclone foraging: each manta swims toward the one in front of it forming spirally move towards the food, where this movement behavior may be extended for n-D space. So, the mathematical equation can be defined as:

\[ x_i^d(t + 1) = x_i^b(t) + r \cdot (x_{i-1}^d(t) - x_i^d(t)) + \beta \cdot (x_i^b(t) - x_i^d(t)) \quad i = 1 \] (4)

\[ x_i^d(t + 1) = x_i^b(t) + r \cdot (x_{i-1}^d(t) - x_i^d(t)) + \beta \cdot (x_i^b(t) - x_i^d(t)) \quad i = 2, ..., N \] (5)

\[ \beta = 2e^{\frac{r-1}{r}} \cdot \sin(2\pi r \cdot t) \] (6)
Where $T$ represents the maximum iteration, $\beta$ is the weight coefficient and $r_1$ a random number in the range $[0, 1]$.

Another mechanism focuses on exploring the search space by making each search for a new position from the current best position by assigning random position as follows:

$$x_{rand}^d = Lb^d + r \cdot (Ub^d - Lb^d)$$  \hspace{1cm} (7)

$$x_i^d(t + 1) = x_{rand}^d(t) + r \cdot (x_{brand}^d(t) - x_i^d(t)) + \beta \cdot (x_{rand}^d(t) - x_i^d(t)) \quad i = 1$$  \hspace{1cm} (8)

$$x_i^d(t + 1) = x_{rand}^d(t) + r \cdot (x_{i-1}^d(t) - x_i^d(t)) + \beta \cdot (x_{brand}^d(t) - x_i^d(t)) \quad i = 2, ..., N$$  \hspace{1cm} (9)

Where $L$ and $U$ represent the lower and upper boundaries for search space and $x_{rand}^d$ is the random position.

3. Somersault foraging in this strategy it swims in pivot way and update their positions always around the best position found so far. Where the mathematical model can be shown as follow:

$$x_i^d(t + 1) = x_i^d(t) + S \cdot (r_2 \cdot x_{2}^d - r_3 \cdot x_i^d(t)), \ i = 1, ..., N$$  \hspace{1cm} (10)

where $S$ is the somersault factor, $r_2$ and $r_3$ are random numbers in $[0, 1]$.

**Figure 1.** Show the strategies of the manta (a) chain foraging (b) cyclone foraging and (c) Somersault foraging.
4. Proposed Method

Generally, the CNN model consists of convolutional layer, pooling layer and at the end fully connected layer. The selection of the hyper-parameter of the CNN model is very difficult for non-expert person. In this paper, we propose a method to select the specific hyper-parameters and improve the accuracy obtained by the model. So, the goal is to find the optimum solution for available hyper-parameters. In MRFO, the lower and upper boundaries represent the range of the hyper-parameters that form the search space, the best position represent the optimum hyper-parameters and \( f(x) \) is the function value returned by CNN model to MRFO algorithm. A noteworthy proposed algorithm tailored to any CNN architecture. The proposed method starts by setting randomly lower and upper boundaries and max number of iterations. After that MRFO algorithm firstly initializes the hyper-parameters, which represents the current position. Next, it begins to update its position depending on the previous strategies to get the best position (optimum hyper-parameters) with high accuracy and less loss. where accuracy and loss calculated simply from training CNN model.

```plaintext
WHILE step criterion is not satisfied do
    FOR i=1 TO N DO
        IF rand < 0.5 THEN //Cyclo foraging
            IF \( u \cdot T_{max} < rand \) THEN
                \[ x_{new} = x_i + rand \cdot (x_u - x_i) \]
            ELSE
                \[ x_{new} = x_i + r \cdot (x_{best} - x_i) + \beta \cdot (x_{rand} - x_i) \] \( i = 1 \)
                \[ x_{new} = x_{best} + r \cdot (x_{best} - x_i) + \beta \cdot (x_{rand} - x_i) \] \( i = 2, \cdots, N \)
            END IF.
        ELSE //Chain foraging
            \[ x_{new} = x_i + r \cdot (x_{best} - x_i) + \beta \cdot (x_{best} - x_i) \] \( i = 1 \)
            \[ x_{new} = x_i + r \cdot (x_{i-1} - x_i) + \beta \cdot (x_{best} - x_i) \] \( i = 2, \cdots, N \)
        END IF.
    END FOR.
    Compute the fitness of each individual \( f(x_i(t+1)) \). IF \( f(x_i(t+1)) < f(x_{best}) \)
    THEN \( x_{best} = x_i(t+1) \)
    END IF.
    //Sommersault foraging
    FOR i=1 TO N DO
        \[ x_i(t+1) = x_i(t) + S \cdot (r_i \cdot x_{best} - r_i \cdot x_i(t)) \]
    Compute the fitness of each individual \( f(x_i(t+1)) \). IF \( f(x_i(t+1)) < f(x_{best}) \)
    THEN \( x_{best} = x_i(t+1) \)
    END FOR.
END WHILE.
Return the best solution found so far: \( x_{best} \).
```

Figure 2. Pseudocode of MRFO algorithm
5. Experiments and Results
In this study, an experiment is studied with cifar10 dataset. Figure (3) shows an example of the dataset.

Figure 3. Cifar10 dataset example

5.1 Experiment 1 using CIFAR-10 dataset with the proposed model
The cifar-10 dataset contains 60000 32X32 RGB images belonging to 10 classes (6000 images per class) and divided into 50000 images for training portion and 10000 images for testing. It is used as a benchmark dataset for image classification. In the experiment, optimization is performed with MRFO algorithm to select the hyper-parameters. The proposed model used is shown below:

1- Convolution layer with 32 and 3x3 filter.
2- Convolution layer with \(c_2\) and 3x3 filter. \(c_2\) range (27-37)
3- Max pooling layer with 2x2 filter.
4- Dropout layer with 0.25 to avoid overfitting
5- Convolution layer with \(c_3\) and \(s_f\ \times \ s_f\) filter. \(c_3\) = (60-72) \(s_f\) = (2-9)
6- Max pooling layer with 2x2 filter.
7- Dropout layer with 0.25 to avoid overfitting
8- FC layer with \(d\) neurons. \(d\) = (192-1024)
9- Dropout layer with 0.5 to avoid overfitting
10- FC layer with ten neurons.
11- Learning rate = \(k\)
12- batch size = p rang (8-86)

The hyper-parameters configure changes every five epochs by MRFO until getting optimum configuration.

5.2 Experiment 2 Applied MRFO for Lenet5 model
In this experiment, MRFO algorithm is applied for Lenet5 model with cifar10 and Mnist dataset and the changes in the models are shown in Table 1 below:
Table 1. Shows the change in hyper-parameters values for Lenet5 model

| Hyper-parameters   | baseline | range    | New value |
|--------------------|----------|----------|-----------|
| Number of filters in C2 | 16       | 13-30    | 21        |
| Number of neurons in FC2 | 84       | 64-84    | 83        |
| Learning rate      | 0.001    | 0.0001-0.005 | 0.003    |
| batch size         | 128      | 32-128   | 92        |
| Activation function| tanh     | Tanh, ReLU | ReLU      |

Figure 4 and Figure 5 shows a comparison between Lenet5 after (LenMRFO) and before (Lenet5) using the MRFO optimization method algorithm. It can be seen that there was an improvement in terms of increasing accuracy and decreasing the loss values.

Figure 4. Shows the accuracy before and after using MRFO optimization method for Lenet5 model with cifar10 dataset
Figure 5. Shows the loss values before and after using the MRFO optimization method for Lenet5 model with cifar10 dataset

The accuracy and loss are studied for Mnist dataset, the results are shown in the figures 6 and 7:

Figure 6. Shows the accuracy before and after using MRFO optimization method for Lenet5 model with Mnist dataset
5.3 Experiment 3
In the third experiment AlexNet model is optimized by the MRFO algorithm with cifar10 dataset and compared its results to the original model. Dropout layer added after both first and second fully connected layers. Table (2) shows the changes in the hyper-parameters of AlexNet model after applying the optimization algorithm with the upper and lower boundaries used for each one.

| Hyper-parameters | baseline | range          | New value |
|------------------|----------|----------------|-----------|
| Strides in C1    | 4        | 2-11           | 2         |
| Kernel size in C1| 11       | 2-15           | 5         |
| Number of filters in C2 | 256   | 128-260       | 189       |
| Kernel size in C2 | 5       | 2-11           | 10        |
| Number of filters in C4 | 384  | 264-396       | 325       |
| Number of neurons in | 4096 | 192-4096     | 847       |
| Learning rate    | 0.001    | 0.0001-0.005  | 0.0002    |
| batch size       | 128      | 32-128        | 60        |
| Activation function | tanh | Tanh, ReLU    | ReLU      |
Then optimized model is trained with cifar10 dataset for 15 epochs and from the results we can see that the optimized model has higher accuracy and fine loss comparing with same model before optimization as shown in the figures 8 and 9:

![Accuracy Comparison](image1)

**Figure 8.** Shows the accuracy before and after using MRFO optimization method for AlexNet model with cifar10 dataset

![Loss Comparison](image2)

**Figure 9.** Shows the loss values before and after using the MRFO optimization method for AlexNet model with cifar10 dataset

6. **Conclusion and Future work**

This paper presented MRFO as an optimization method to select the hyper-parameters values of CNN. The proposed method has been studied and experimented on Lenet5 and AlexNet models using cifar-10 dataset. The obtained results showed that using MRFO has improved the performance in terms of increasing the accuracy and decreasing the loss values. This research can be extended in terms of exploring this method in more CNN models like VGG-16.
References

[1] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. Paper presented at the 2017 International Conference on Engineering and Technology (ICET).

[2] Albeahdili, H. M., Han, T., & Islam, N. E. (2015). Hybrid algorithm for the optimization of training convolutional neural network. International Journal of Advanced Computer Science & Applications, 1(6), 79-85.

[3] Ismail, A., Ahmad, S. A., Soh, A. C., Hassan, K., & Harith, H. H. (2019). Improving Convolutional Neural Network (CNN) architecture (miniVGGNet) with Batch Normalization and Learning Rate Decay Factor for Image Classification. International Journal of Integrated Engineering, 11(4).

[4] Junior, F. E. F., & Yen, G. G. (2019). Particle swarm optimization of deep neural networks architectures for image classification. Swarm and Evolutionary Computation, 49, 62-74.

[5] Lorenzo, P. R., Nalepa, J., Kawulok, M., Ramos, L. S., & Pastor, J. R. (2017). Particle swarm optimization for hyper-parameter selection in deep neural networks. Paper presented at the Proceedings of the genetic and evolutionary computation conference.

[6] Serizawa, T., & Fujita, H. (2020). Optimization of Convolutional Neural Network Using the Linearly Decreasing Weight Particle Swarm Optimization. arXiv preprint arXiv:2001.05670.

[7] Smith, L. N. (2017). Cyclical learning rates for training neural networks. Paper presented at the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV).

[8] Takahashi, R., Matsubara, T., & Uehara, K. (2018). A novel weight-shared multi-stage CNN for scale robustness. IEEE Transactions on Circuits and Systems for Video Technology, 29(4), 1090-1101.

[9] Zhao, W., Zhang, Z., & Wang, L. (2020). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. Engineering Applications of Artificial Intelligence, 87, 103300.