Artificial Bee Colony Algorithm Based Hyper-Parameter Optimization for Convolutional Neural Networks

1st Koray Ozdemir
Dept. of Computer Engineering
Institute of Science
Yalova University
Yalova, Turkey
korayozdemir34@gmail.com

2nd Yunus Ozen
Dept. of Computer Engineering
Engineering Faculty
Yalova University
Yalova, Turkey
yunus.ozen@yalova.edu.tr

3rd Adem Tuncer
Dept. of Computer Engineering
Engineering Faculty
Yalova University
Yalova, Turkey
adem.tuncer@yalova.edu.tr

***Abstract*—In recent years, Deep Learning has become a field that researchers are particularly interested in. The Convolutional Neural Network (CNN) is a type of multi-layered artificial neural network mostly used in the analysis, recognition, and classification of images and videos. The performance of CNN models is usually based on custom model architectures, thus several hyper-parameter values in a CNN are manually selected mostly. However, different combinations of hyper-parameters and models need to be used to achieve better performance results. Determination of optimum values of hyper-parameters is also an optimization problem. The meta-heuristic optimization techniques are able to solve such problems. In this paper, we propose to use the Artificial Bee Colony (ABC) algorithm which is a meta-heuristic approach to automatically determine the optimum architecture of a CNN by means of hyper-parameters. The most effective hyper-parameters in the performance of CNN models have been optimized, which are the number of layers, the number and size of filters, activation function, batch size, learning rate, optimizer, and dropout rate. We have evaluated our optimized architecture using the well-known Fashion-MNIST dataset. The results demonstrate that the proposed model using ABC improves the performance of a CNN model.**

Keywords—convolutional neural networks, artificial bee colony, hyper-parameters, optimization, deep learning

I. INTRODUCTION

Artificial Neural Networks (ANNs), one of the common machine learning methods, are computing systems created by imitating the structure of biological neural networks. They are based on the information processing and analysis processes of the human brain. ANNs are widely used in solving real world problems such as speech recognition, visual recognition, and natural language processing [1]. The concept of Deep Neural Network (DNN) has emerged by increasing the number of hidden layers in the structure of ANNs. DNNs, unlike traditional ANNs, consist of more than two hidden layers connected with each other [2]. Its advanced structure produces high performance results in fields such as image recognition and speech recognition compared to traditional ANNs [3], [4].

Convolutional Neural Networks (CNNs), a special type of DNNs, were firstly introduced in 1998 for document recognition tasks [5]. They have been widely used in fields such as handwriting recognition [6] and image classification [7]. In addition, the use of CNNs in robotics has significantly increased in recent years due to the advances in processing technologies [8]. In order to increase the performance of CNNs, studies are generally carried out on the training phase [9]. In addition to the training phase, hyper-parameter optimization also plays an important role in improving the performance of CNNs. The hyper-parameters such as filter size, number of hidden layers, learning rate, and activation function are needed to be optimized. The correct tuning of the hyper-parameters has a direct impact on the performance of a CNN. These parameters are set manually or used by methods such as grid search or random search. In the manual search method, it is necessary to rely on the intuition and experience of the researcher in order to set the hyper-parameters correctly. However, the training process needs to be repeated with each new set of hyper-parameters, and this is quite laborious and time-consuming. The grid search method is one of the traditional methods used in setting hyper-parameters. In a study conducted by Liashchynskyi [10], it has been stated that the grid search method is not effective in high dimensional spaces and its positive effect on network performance is limited. The random search method, which is another hyper-parameter setting method, is based on running all possible combinations. Bergstra and Bengio [11] compared manual search, grid search, and random search methods and they found that the random search method alone gives more successful results than others. However, the random search method is non-adaptive. In other words, the previously tried hyper-parameters are not taken into account when trying new parameters. For this reason, it has been observed in the study that the combination of manual search and random search methods is more successful than the random search method. The use of meta-heuristic approaches in CNNs for hyper-parameter selection is a potential field of study when compared to traditional search methods. Young et al. [12] used genetic algorithms (GA) in the hyper-parameter optimization of CNNs. They used the CIFAR-10 dataset for experiments and stated that the GA approach was more effective than random search.

In this study, the Artificial Bee Colony (ABC) algorithm, one of the nature-inspired meta-heuristics approaches, has been used in the hyper-parameter optimization of CNNs.

II. BACKGROUND

A. Artificial Bee Colony Algorithm

ABC algorithm was introduced by Dervis Karaboga in 2005 for the purpose of optimizing numerical problems [13]. It is a swarm-based meta-heuristic algorithm based on imitating the foraging behaviors of honey bees in nature. ABC algorithm basically consists of three types of bees: employed bees, onlooker bees, and scout bees [14]. Employed bees

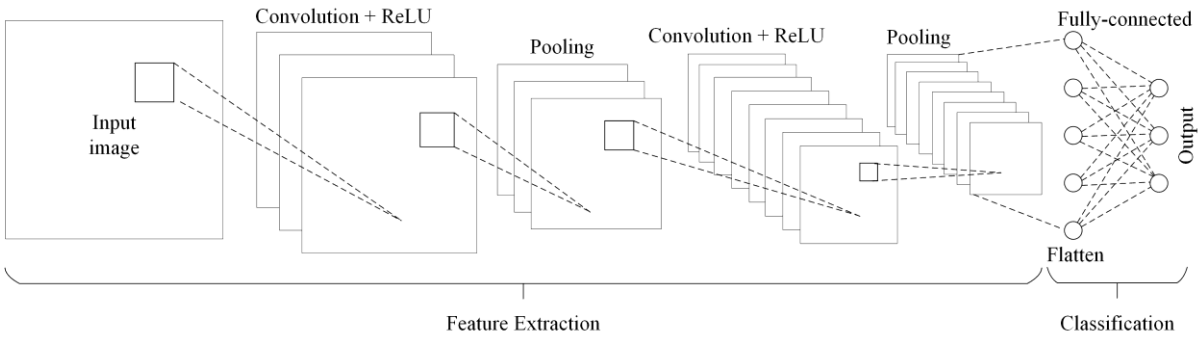

Fig. 1. The basic architecture of CNN

search to find rich food sources. Rich food sources represent the best solutions to the given problem. Then, the employed bee shares information regarding the food source with the onlooker bee. The onlooker bee can choose a source using this information. The scout bees are responsible for finding new food sources instead of abandoned food sources. In the ABC algorithm, all parameters to be optimized of the problem are represented vector and is tried to find the best parameter vector which minimizes an objective function. Each of the parameter vectors is named a food source. The algorithm starts with the initial random values of the parameter. In each iteration, while poor food sources are abandoned, rich food sources are reached by using the neighboring solution mechanism. Thus, food sources that represent the best solution to the given problem are found.

The initial food sources, x_i, are generated randomly in the ABC algorithm using (1).

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \qquad (1)$$

where i represents ith food source which is a candidate solution and j represents jth parameter to be optimized, x_j^{min} and x_j^{max} are lower and upper bounds of the jth parameter. A random number between 0 and 1 is generated by $rand(0,1)$.

The employed bee searches the neighboring food sources, v_i, according to (2). The information on food sources is kept by employed bees and passed on to onlooker bees.

$$v_{ij} = x_{ij} + rand(-1,1)(x_{ij} - x_{kj}) \qquad (2)$$

where k and j are randomly chosen numbers and they have to be different. $rand(-1,1)$ enables neighboring food sources to be found and represents comparing two different neighboring food sources [14].

An onlooker bee selects a food source according to its probability value, p_i, associated with its quality calculated by (3).

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad (3)$$

where fit_i is the fitness of the food source i. SN is the number of food sources and it equals number of employed bees.

If the food source cannot be further improved as a certain number of cycles, which is called limit, the food source is abandoned. Instead of that food source, a new food source is generated by the scout bee using (1). When a predetermined maximum number of cycles is reached, the algorithm is terminated. The pseudo-code of the ABC algorithm [13] is shown in Algorithm 1.

Algorithm 1 The pseudo-code of the ABC algorithm

Initialize population with random solutions x_i by (1)
Evaluate the fitness of the population
cycle = 1
Repeat
  Generate new solutions v_i by (2) and evaluate the fitness
  Apply the greedy selection between x_i and v_i
  Select a solution depending on the probability p_i by (3)
  Generate new solution v_i and evaluate the fitness
  Apply the greedy selection between x_i and v_i
  If employed bee becomes scout
    Then replace it with a new produced solution by (1)
  Memorize the best solution obtained so far
  cycle = cycle + 1
Until Maximum number of cycles is reached

B. Convolutional Neural Networks

CNN is a subclass of DNNs widely used in image and video recognition, recommendation systems, image classification, and natural language processing. CNNs, unlike traditional ANNs, show high performance especially in the classification of high dimensional image data [15]. One of the advantages of CNNs is that they can learn directly from data with automatic feature extraction, eliminating manual feature extraction [16]. A CNN consists of several layers:

• Convolutional layer: In this layer, convolution operations are applied using filters in various sizes and types to the input for extracting features.

• Non-linearity layer: This layer takes the feature map produced by the convolutional layer and applies an activation function to bring non-linearity into the output. ReLU (rectified linear unit) is generally used as the activation function in CNNs.

• Pooling layer: This layer provides a reduction of the spatial dimension and the number of parameters.

• Flattening layer: The layer where the data is flattened and prepared for the fully connected layer.

• Fully connected layer: This layer is an ANN in which each unit in layers is connected to every unit in the other layers. The training phase required to classify the features obtained in the previous layers into various classes is performed in this layer.

In Fig. 1, the basic architecture of CNN in which convolution, ReLU activation function, and pooling process are applied twice, respectively, are given.

III. ABC Algorithm Based Hyper-Parameter Optimization

A. Problem statement

Hyper-parameters such as the number of convolution layers and pooling layers, the number and size of filters, the number of fully connected layers, and the number of neurons in each layer directly affect the performance of CNNs. However, the search field consisting of values defined for all hyper-parameters will be very large since each hyper-parameter has different values separately. In this respect, trying all options will be very costly in terms of computation time. The usage of the manual search method decreases over time as it is based on the intuition of the researchers and requires a considerable amount of time. The grid search method, which is one of the alternative methods, does not give effective results in high dimensional spaces, and the computational cost is high. A disadvantage of the random search method, which provides a higher performance compared to the other two methods, is its non-adaptive structure [12]. The transferability of the selected parameters is as essential as the hyper-parameter selection method. Hyper-parameter fine-tuning for simple architectures generally does not have the same effect as for complex architectures. Similarly, success in an image dataset may not be seen in other datasets [17]. The shortcomings of the current hyper-parameter selection methods have increased the use of meta-heuristic algorithms in this area. For this reason, the ABC algorithm has been used in this study for determining the optimum hyper-parameter values.

B. The dataset and hyper-parameters

Fashion-MNIST [18] dataset was used in this study. In the dataset, there are 28×28 grayscale images of 70000 fashion products belonging to 10 classes. Each class has 7000 images. 60000 of the images are selected as training data and 10000 of them as test data. Fashion-MNIST dataset has the same structure and image size as the original MNIST dataset.

A food source in the ABC algorithm consists of hyper-parameters used for a CNN architecture which are in order of the number of convolutional layers, number of filters per convolutional layer, number of fully connected layers, filter size, dropout rate, the type of activation function, optimizer, batch size, and the size of the fully connected layer. Candidate solutions have been generated by using (1), considering the upper and lower limits of each distinct hyper-parameter. Hyper-parameters and their boundaries are shown in Table I. The categorical cross-entropy has been used as a cost function. The pooling size is set to 2×2 fixed and is not included in the optimization. In order to reduce the search space and thus reduce computational costs, the upper bounds of the hyper-parameters were predetermined. In the ABC algorithm, the number of food sources, limit, and maximum number of cycles were set to 100, 100, and 100, respectively. Dimension of the food source is set to 9, which equals hyper-parameters to be optimized. The number of epochs for the training of candidate CNN models was also set to 15.

TABLE I. Hyper-Parameters and Value Ranges to be Optimized

Hyper-parameter	Value ranges
Number of convolutional layers	[1, 6]
Number of filters per conv. layer	[1, 128]
Number of fully connected layers	[1, 3]
Filter size	[1, 8]
Dropout rate	[0, 0.5]
Activation function	ReLU, LReLU, ELU
Optimizer	Adam, Adadelta, SGD
Batch size	32, 64, 128, 256
The size of fully connected layer	[16, 1024]

IV. Experimental Results

The experiments were performed using a computer with an i7 processor and an Nvidia 1050TI GPU. According to the results, it has been observed that the network architecture that reaches the highest accuracy values has three convolution layers and one fully connected layer. The most successful results have been obtained when filter size is $3x3$ and the number of units in the fully connected layer is 51. As an activation function, it has been seen that ReLU gives better performance results than other activation functions. Adam optimizer has better results compared to the other two optimizers. Optimum hyper-parameter values obtained from the experiments are shown in Table II.

TABLE II. Optimum Hyper-Parameter Values Obtained

Hyper-parameter	Value
Number of convolutional layers	3
Filter size for conv. layer 1	3
Number of filters for conv. layer 1	102
Filter size for conv. layer 2	3
Number of filters for conv. layer 2	126
Filter size for conv. layer 3	3
Number of filters for conv. layer 3	29
Dropout rate	0.477
Activation function	ReLU
Optimizer	Adam
Batch size	64
Number of fully connected layers	1
Number of units in the fully connected layer	51

The results have been compared with hyper-parameter optimization studies using the Fashion-MNIST dataset and different methods. Comparisons have been made on accuracy values. The accuracy values of other studies and our study used Fashion-MNIST dataset are shown in Table III.

TABLE III. ACCURACY (%) RESULTS OF THE PERFORMANCE COMPARISON AMONG RELATED WORKS

Study	Method	Accuracy
Dufourq (2017) [19]	GA	90.60
Sun (2017) [20]	GA	92.72
Ma (2018) [21]	GA	94.59
Assunçao (2018) [22]	GA	95.26
Our study	**ABC**	**93.46**

V. CONCLUSION

Hyper-parameter fine-tuning plays an important role in improving the performance of CNNs. Manual search, grid search, and random search methods, which are the traditional methods used, have some deficiencies. For this reason, the use of meta-heuristic approaches in hyper-parameter optimization has increased and these methods have shown higher success than traditional methods. In this study, the ABC algorithm, one of the swarm based algorithms, has been used in hyper-parameter optimization of CNNs. Optimized hyper-parameters are the number of convolutional layers, number of filters per convolutional layer, number of fully connected layers, Filter size, dropout rate, the type of activation function, the type of optimizer, batch size, and the size of fully connected layers. Accuracy values obtained were compared with other studies using the Fashion-MNIST dataset under similar test conditions. The accuracy values of our study give similar results to other studies. As a result, it has been seen that the ABC algorithm achieves high success in hyper-parameter optimization of CNNs.

REFERENCES

[1] B. Yegnanarayana, "Artificial neural networks," PHI Learning Pvt. Ltd.. nx, 2009.

[2] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.

[3] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'17, (New York, NY, USA), pp. 497–504, ACM, 2017

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Processing Magazine, vol. 29, pp. 82–97, 2012.

[5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition" Proceedings of the IEEE, vol. 86, pp. 2278–2324, 1998.

[6] A. Baldominos, Y. Saez, and P. Isasi, "Evolutionary convolutional neural networks: An application to handwriting recognition," Neurocomputing, vol. 283, pp. 38–52, 2018.

[7] N. Jmour, S. Zayen, and A. Abdelkrim, "Convolutional neural networks for image classification," In 2018 International Conference on Advanced Systems and Electric Technologies, pp. 397–402, March 2018.

[8] S. Kumra, and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 769–776, IEEE, 2017.

[9] Y. Chhabra, S. Varshney, and Ankita, "Hybrid particle swarm training for convolution neural network (cnn)," In Tenth International Conference on Contemporary Computing (IC3), pp. 1–3, Aug 2017.

[10] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for nas," arXiv preprint arXiv:1912.06059, 2019.

[11] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," J. Mach. Learn. Res., vol. 13, pp. 281–305, Feb. 2012.

[12] S.R. Young, D.C. Rose, T.P. Karnowski, S.H. Lim, and R.M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," In Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, MLHPC'15, (New York, NY, USA), pp. 4:1–4:5, ACM, 2015.

[13] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical ReportTR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[14] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," Applied mathematics and computation, 214(1), pp. 108–132, 2009.

[15] S. Albawi, T.A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," In International Conference on Engineering and Technology (ICET), pp. 1–6, IEEE, 2017.

[16] M. Sardogan, A. Tuncer, and Y. Ozen, "Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm," In 3rd International Conference on Computer Science and Engineering (UBMK), pp. 382–385, 2018.

[17] T.M. Breuel, "The effects of hyperparameters on SGD training of neural networks," arXiv preprint arXiv:1508.02788, 2015.

[18] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," arXiv preprint arXiv:1708.07747, 2017.

[19] E. Dufourq and A.B. Bruce, "EDEN: Evolutionary deep networks for efficient machine learning," Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech), IEEE, 2017.

[20] Y. Sun, X. Bing and M. Zhang, "Evolving deep convolutional neural networks for image classification," arXiv preprint arXiv: 1710.10741, 2017.

[21] B. Ma and Y. Xia, "Autonomous Deep Learning: A Genetic DCNN Designer for Image Classification," arXiv preprint arXiv:1807.00284, 2018.

[22] F. Assunçao, N. Lourenço, P. Machado and B. Ribeiro. "DENSER: Deep Evolutionary Network Structured Representation," Genetic Programming and Evolvable Machines, pp. 1–31, 2018.