Improving The Performance of Convolutional Neural Networks using Evolutionary Computing

Abdullah Farhan Mahdi
Department of Horticulture, faculty of Agriculture, University diyala, Iraq
abdullahmahdi@uodiyala.edu.iq

Abstract

Convolutional Neural Networks (CNNs) have achieved remarkable success with numerous real-world issues in recent years. The structure of these networks is heavily influenced by a number of parameters, such as the number and type of layers, the size and number of cores, and the type of activation function. In this article, genetic algorithms were used to design CNNs structures, because genetic algorithms are able to apply learning in an automatic way. The algorithm was tested on Cifar10 and Cifar100 datasets, compared with three newly designed CNNs, two competitors designed semi-automatic CNN structures, and others designed fully automatic CNN structures. As the results CIFAR10 Classification error (4.3), Parameters number (M = 106) (2.1 M) and Execution-time (day) (40). CIFAR100 Classification error (20.85), Parameters number (M = 106) (5.5 M) and Execution-time (day) (84). The results also show that the parameters' number of the best structure reached using the proposed algorithm is less than in the automatic algorithms that were compared with it (Block-QNN-S and Large-Scale Evolution). This is within an implementation time of the proposed algorithm of 84 days, according to the computational resources.

Keywords: *Deep neural networks, Genetic algorithms, Convolutional neural network, Skip connection.*

تحسين أداء تجميع الشبكات الطبيعية SNFUL باستخدام الحوسبة المسطحة

عبدالله فرحان مهدي
علوم الحاسوب، كلية الزراعة ، جامعة ديالى، بعقوبة، العراق
abdullahmahdi@uodiyala.edu.iq

الملخص

حققت الشبكات العصبية التلافيفية (CNNs) نجاحاً ملحوظاً مع العديد من القضايا الواقعية في السنوات الأخيرة. يتأثر هيكل هذه الشبكات بشكل كبير بعدد من المعلمات، مثل عدد ونوع الطبقات، وحجم وعدد النوى، ونوع دالة التنشيط. في هذه المقالة، تم استخدام الخوارزميات الجينية لتصميم هياكلCNN ، نظراً لحقيقة أن الخوارزميات الجينية قادرة على تطبيق التعلم بطريقة آلية. تم اختبار الخوارزمية المقترحة على مجموعات البيانات cifar100 وcifar10، وتمت مقارنة النتائج بمجموعة من الخوارزميات التي أظهرت أداءً واعداً في هذا المجال.

الكلمات المفتاحية:الشبكات العصبية العميقة ;الخوارزميات الجينية; الشبكة العصبية التلافيفين ;تخطي الاتصال.

Introduction

Computer vision is an area of modern computing science and a form of artificial intelligence. This technology helps to see the world and analyze visual data to make decisions and gain an understanding of the environment and the world, as well as to identify and process objects such as images and videos in the same way that humans do. Until recently, computer vision operated in a limited capacity, but, the main driving forces which led to the development of computer vision has been the sheer volume of digital data we produce today.

Many AI and machine learning tasks, including image recognition, speech recognition, and reinforcement learning tasks, have demonstrated that deep learning, which employs deep neural networks as a model, performs well. In particular, in the past few years, convolutional neural networks (CNNs) have seen great success in the field of computer vision and specifically in pattern recognition tasks [1].

CNNs provide a very high degree of accuracy compared to other machine learning methods, as these networks are distinguished in their ability to extract features automatically without human intervention and use relatively little initial processing compared to other image classification algorithms.

The performance of CNNs depends greatly on their structure, usually, a deeper structure (the more layers) will produce a better level of accuracy, however, there are some cases the deeper the structure will produce a worse level of accuracy, so defining the CNN structure is a matter extremely important [2].

These networks' structure is determined by a number of parameters known as hyper-parameters, such as the type and number of layers, core size, and activation function. The traditional search methods that are commonly used to determine the values of these parameters, such as Grid [3], Manual [4], and Random Search [5], require a large computational burden due to the large search areas. It also requires extensive knowledge of CNNs and the problem area being worked on, and this knowledge will not be available to all people.

On the other hand, nowadays evolutionary computing is increasingly being used to solve optimization problems including determining the optimal parameters of a function [6], [7]. Based on this, and since the design of the CNN structure is closely related to the selection of many parameters, work began on the use of evolutionary computing techniques (such as genetic algorithms, swarm algorithms, etc.) in determining the optimal parameters in the CNN structure [8].

CNNs for computer vision tasks face several challenges that hinder their deployment in real-world applications, the most important of which are model size, runtime memory, and the number of computations [1]. CNN have recently gained a lot of traction and have been applied in wide fields [9]-[11]. The most recent hand-planned CNN networks, ResNet [12] and DenseNet [13], have achieved remarkable model recognition success, and then multiple algorithms have been proposed to design CNN structures, the most important of which is Genetic CNN [14], Large-Scale Evolution [15], CGP-CNN [16].

This research derives its importance from the emergence of convolutional neural networks as leading models for model recognition tasks, in addition to the effectiveness of genetic algorithms in solving complex problems with a large search space. The goal of this paper is to make an algorithm to discover the best structure of CNNs using genetic algorithms (GAs) according to the task concerned in a completely automatic manner, that does not require any manual intervention during the developmental research stages, taking into account the limited computational resources, where the CNNs obtained after the end of the evolution process can be used directly in data processing, without the need for additional refinement like adding more layers of convolution or pooling. Moreover, the proposed algorithm can be straightforwardly utilized by different researchers who don't have to do any preprocessing, for example, giving a physically preset network.

Convolutional neural networks

Convolutional neural networks are a unique type of feed-forward neural networks that are derived from biological processes in the visual lobe. They are regarded as a potential solution to a number of issues related to computer vision and artificial intelligence [1], [17],

[18]. LENET is one of the first CNNs that revolutionized deep learning, and it was mainly used for character recognition tasks, and it was named LENET-5 [19], [20].

In recent years, CNNs have gained great importance. As ResNet [12] and DenseNet [13] are the latest hand-designed CNNs that have achieved great success in pattern recognition. Then, multiple algorithms are proposed to design CNN structures depending on if it is required pre-processing or post-processing of the CNN structure. Accordingly, based on how the CNN architecture parameters are determined, these algorithms can be classified into two distinct groups: semi-automatic CNN architecture design algorithms (Genetic CNN and Block-QNN-S [14]) and automatic design algorithms (Large-Scale Evolution [15] and CGP-CNN [16]).

The structure of a CNN network is typically consisted of a number of distinct layers, each of which has a distinct function. These layers of CNNs are classified into four layers, which are [8]:

2.1 Convolutional layer

It is the backbone of CNN and its name comes from the mathematical folding or convolution process, where the feature map that reflects the weights of each filter's response to a particular pattern in the image is the output of this process. Additionally, the filter weights are automatically calculated during the network training process [19]. Figure 1 shows an example of the convolution process, assuming that the input image's size is 4 x 3 and the convolution kernel's size is 2 x 2, so that the convolution kernel is overlaid on top of the input image and the numerical product is calculated between the numbers in the same location in the kernel and the inputs.

The feature map consists of several channels, the dimensions of which are related to the input matrix dimensions (considering that the input of these networks is a collection of images), and the dimensions of the filter (considering that some parameters must be specified before the network training process such as the filters' number and size, Network structure, etc.) in addition to the factors: stride, which is the elements' number that the filter is moved by after each operation and Padding, which is the process of zero padding is to add zeros around the edges of the input image matrix, which helps pass the filter on the edges of the image better. The zero-padding operation is useful for controlling the size of the feature maps generated by convolution. It is usually used to make the output size of the convolutional layer (feature maps' size) equal to the input size.

(a) (b)

Figure 1. Convolution operation (a) 2 x 2 kernel and (b) convolution input and output.

2.2. Activation layer

After the convolution process is completed, the feature map is inserted into the activation layer, where the activation function is applied to each neuron, which is equivalent to an element of the feature map. The most important activation function used in this type of network is the ReLU function, which has proven its effectiveness compared to other functions [21].

2.3. Pooling layer

This layer is not required to be present in the network's design; rather, if it does exist, it will be placed after each convolutional layer. It aims to make the feature map smaller (reduce its size) by reducing each size-specific group of input neurons down to a single one, and this size is characterized by a little window inside the network design. The reduction is done in more ways than one, the most significant of which is max-pooling, where every window is matched with one element that represents the window's highest value [22].

The output of the pooling process is a feature map that has a similar depth but different widths and heights. Therefore, the pooling process has many benefits, the most prominent of which are: 1) Reduce the dimensions of the feature map and the number of computation variables and computations in the network and 2) It makes the network resistant to minor shifts or the input matrix's distortion.

Figure 2 shows an example of the Max-pooling process, where the matrix on the left represents the feature map (input of the pooling layer) and the matrix on the right is the output of the pooling process through a 2 x 2 window with a step of 2.

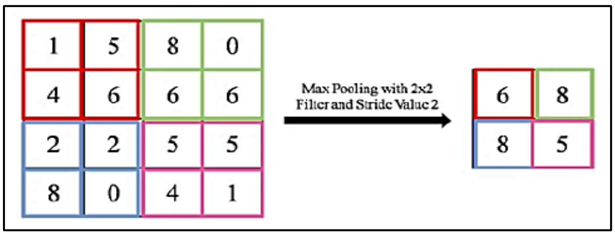


Figure 2. Max-pooling process2.4. Fully connected layer

This layer is the last in the CNN, and it is of the multi-layer perceptron type, in which the neurons are fully connected with all the previous layer's nodes, and the final step in the classification process is taken. Its input is a vector formed from the feature map after the pooling process, and its output is a vector that expresses the row (class) to which the feature map belongs [23].

Genetic algorithms (GA)

GAs were inspired by theory of Darwin, which said that "survival of the fittest", by generating new individuals (chromosomes) via process like crossover and mutation. This indicates that the fittest individual has a greater chance of survival and mating. Therefore, the next generation's population will be stronger because it is made up of strong individuals, that is, the solution changes and evolves with each generation.

The initial set of random solutions is the starting point for GA. These solutions (individuals) are then coded according to the current problem, and a fitness function is used to determine how good each individual is. GA mainly depends on three factors as follow: 1) selection: The procedure of finding the community's "best" parents with whom to mate, the "best" is selected based on the current problem, 2) crossover operator: It is the process of merging parents (chromosomes) to form new offspring by exchanging parts of their genes. The new offspring are more likely to have good parts of their parents and thus perform better compared to their predecessors and 3) mutation operator: takes one chromosome and changes some of its genes to create a new chromosome [24].

For basic stages of a genetic algorithm, the algorithm begins by characterizing the problem in order to represent the chromosomes representing the solutions by one of the coding methods, then a random set of candidate solutions is generated, each of which is referred to as a chromosome, and to the whole set as a population, with reference to Initial population of the first generation.

In each generation, the individuals of the population are selected and combined in an effort to 'reproduce' the chromosomes with a higher degree of fitness; This process is called crossover. Our choice of crossover mainly relies upon the sort of coding utilized; this can occasionally be very complex but usually improves the performance of the genetic algorithm. However, whether or not this sort of crossover is superior to others stays an open inquiry.

After a certain number of new chromosomes have been generated in this way, a subset of them (those with the highest fitness function) replaces an equal portion of the existing population (those with low fitness function). This process is then repeated generation after generation, with the population remaining at a constant size, until a certain number of generations have passed, or a chromosome scores higher than a predetermined value for the fitness function; Then the algorithm returns the best chromosome (with the highest value of the fitness function) as the optimal solution [25].

3.1. Execution stages of the algorithm

The algorithm's general framework is made up of the following steps:

1) The population is initialized based on the individual coding strategy where the individual represents a proposed CNN structure to solve the classification problem.

2) The process of evolution continues until the predetermined cessation criterion is met, which is the maximum number of generations in this work: First, the fitness scale, which is the accuracy of the individual's classification on the database, is used to evaluate each individual. Then, the selection of solutions (parents) using the principle of binary tournament selection. Next, the creation of new offspring using genetic factors crossover and mutation. Finally, the selection of the best individuals from the community of parents and the community of new offspring to build the population for the next generation and take part in evolution later on.

3) After the end of the evolution process, the best individual is chosen and then it will be decoded into the CNN that corresponds.

Figure 3 summarizes the implementation stages of the algorithm, and the following is a detailed explanation of each of these stages:

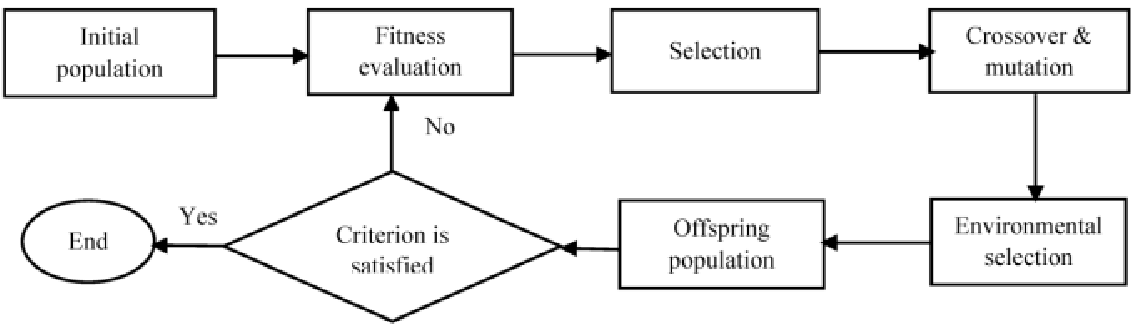


Figure 3. Algorithm execution stages.

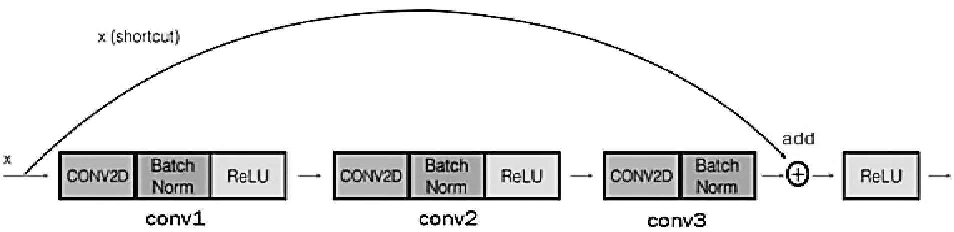
Strategy of Coding the Individual and Forming Population

In the proposed algorithm, the individual represents a proposed CNN network structure to solve the classification problem. The algorithm is based on modern network structures that have been designed and tested by other researchers before, which are the residual networks (ResNets) [12] and the Densely connected networks (DenseNets) [13], where the use of skip connections by these networks revolutionized the field.

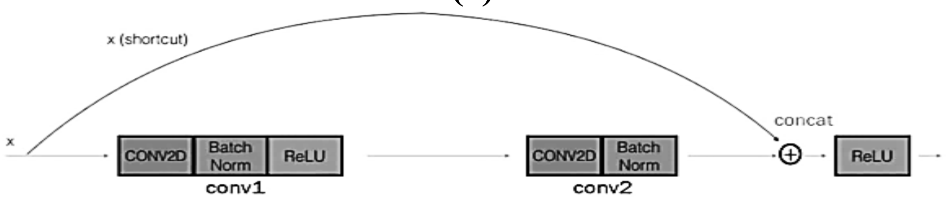
The skip connection is a standard unit in many convolutional network architectures, as it bypasses some layers in a neural network by making the output of one layer as input to subsequent layers (instead of just the next one), thus providing an alternative path for the derivatives of the error function during signal backpropagation (with backpropagation). Skip connections between different non-sequential layers are used in one of two ways: addition as in the remaining networks ResNets and concatenation as in DenseNets.

In ResNet unit (RU), each unit consists of a set of RBs (ResNet Blocks), where the number of these blocks is considered as a hyper-parameter whose value is randomly determined during the execution of the algorithm. Figure 4(a) shows an example of a residual block (RB) consisting of a sequence of three convolution layers and one skip connection.

While in DenseNet unit (DU), each unit consists of a set of DBs (DenseNet Blocks), where the number of these blocks is considered as a hyper-parameter whose value is determined randomly during the execution of the algorithm. Figure 4(b) shows an example of a DB with two convolution layers and one skip connection.



(a)



(b)

Figure 4. ResNet-Block architecture (a) and DenseNet-Block architecture (b)

The results also show that the parameters' number of the best structure reached using the proposed algorithm is less than in the automatic algorithms that were compared with it (Block-QNN-S and Large-Scale Evolution). This is within an implementation time of the proposed algorithm of 84 days, according to the computational resources available to us.

CONCLUSION

In this work, an algorithm is developed to design the CNN architecture using GAs according to the required task in a fully automatic manner. An encryption technique based on complex network blocks with representations of individuals of varying lengths was used to accomplish this objective.

The algorithm was tested on Cifar10 and Cifar100 datasets, compared with three newly designed CNNs, two competitors designed semi-automatic CNN structures, and others designed fully automatic CNN structures. The results showed a promising performance in terms of the parameters' number and the classification error-rate of the proposed algorithm compared with other algorithms.

In this work, GA was used to explore the network structure, and the repair process (correcting invalid individuals) used in the algorithm was very time-consuming because there is no limit to stopping it, as it keeps changing and intersecting until a valid individual is reached. Therefore, it would be a good idea to develop a more efficient way to recover unsuitable individuals without repeated crossovers and mutations. In addition, the computational complexity of the proposed algorithm should be studied and reduced without affecting the classification accuracy.

REFERENCES

[1] Kadhim, Q.K., Altameemi, A., Jasim, S, " Artificial intelligence techniques for colon cancer detection: a review" . J. Yarmouk 21(2),pp. 11–18,2023.

[2] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," presented at the Proc. Genetic and Evol. Comput. Conf., Berlin, Germany, 2017. doi:10.1145/3071178.3071229.

[3] E. Real *et al.*, "Large-Scale Evolution of Image Classifiers," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, Australia, 2017, pp. 2902-2911.

[4] F. Dernoncourt and J. Y. Lee, "Optimizing neural network hyperparameters with Gaussian processes for dialog act classification," in *2016 IEEE Spoken Lang. Technol. Workshop (SLT)*, 2016, pp. 406-413.

doi:10.1109/SLT.2016.7846296.

[5] Kadhim, Q.K., et al, "COVID-19 disease diagnosis using artificial intelligence based on gene expression: a review". Sumer J. Pure Sci. 2(2),pp 88–102 ,2023.

[6] A. I. Bratcu, L. Makdessian, and A. Dolgui, "Minimisation of equipment cost for transfer lines with blocks of parallel tasks," in *Proc. IEEE Int. Symp. Assem. and Task Planning, 2003.*, 2003, pp. 109-114. doi:10.1109/ISATP.2003.1217196.

[7] H. Chehade, A. Dolgui, F. Dugardin, L. Makdessian, and F. Yalaoui, "Multi-Objective Approach for Production Line Equipment Selection," *Manage. Prod. Eng. Rev.*, vol. 3, no. 1, pp. 4-17, 2021, doi:10.2478/v10270-012-0001-5.

[8] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455-5516, 2020, doi:10.1007/s10462-020-09825-6.

[9] G. S. Saragih, Z. Rustam, and J. E. Aurelia, "A hybrid model based on convolutional neural networks and fuzzy kernel k-medoids for lung cancer detection," *Indonesian J. Elec. Eng. Comput. Sci.*, vol. 24, no. 1, pp. 126-133, 2021, doi:10.11591/ijeecs.v24.i1.pp126-133.

[10] U. Dudekula and P. N, "Linear fusion approach to convolutional neural networks for facial emotion recognition," *Indonesian J. Elec. Eng. Comput. Sci.*, vol. 25, no. 3, pp. 1489-1500, 2022, doi:10.11591/ijeecs.v25.i3.pp1489-1500.

[11] S. Belattar, O. Abdoun, and E. k. Haimoudi, "Performance analysis of the application of convolutional neural networks architectures in the agricultural diagnosis," *Indonesian J. Elec. Eng. Comput. Sci.*, vol. 27, no. 1, pp. 156-162, 2022, doi:10.11591/ijeecs.v27.i1.pp156-162.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, 2016, pp. 770-778. doi:10.1109/CVPR.2016.90.

[13] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger" ,Densely Connected Convolutional Networks," in *2017 IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, 2017, pp. 2261-2269. doi:10.1109/CVPR.2017.243.

[14] Z. Zhong, J. Yan, W. Wu, J. Shao, and C. L. Liu, "Practical Block-Wise Neural Network Architecture Generation," in *2018 IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, 2018, pp. 2423-2432. doi:10.1109/CVPR.2018.00257.

[15] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks

for Large-Scale Image Recognition," presented at the 3rd Int. Conf. Learn. Representations (ICLR2015), 2015. https://arxiv.org/abs/1409.1556.

[16] L. Xie and A. Yuille, "Genetic CNN," in *2017 IEEE Int. Conf. Comput. Vision (ICCV)*, 2017, pp. 1388-1397. doi:10.11/09ICCV.2017.154.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi:10.1145/3065386.

[18] Q. K. Kadhim, “Image compression using discrete cosine transform method,” International Journal of Computer Science and Mobile Computing, vol. 5, no. 9, pp. 186–192, 2016.

[19] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. 2010 IEEE Int. Symp. Circuits and Syst.*, 2010, pp. 253-256. doi:10.1109/ISCAS.2010.5537907.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015, doi:10.1038/nature14539.

[21] . Ahmed, S.T., Khadhim, B.J., Kadhim, Q.K,“Cloud services and cloud perspectives: a review“. IOP Conf. Ser. Mater. Sci. Eng. 012078 ,2021. https://doi.org/10.1088/1757-899X/1090/1/012078

[22] C.-Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree," in *Proc. 19th Int. Conf. Artif. Intell. and Statist. (AISTATS)*, 2016, pp. 464-472.

[23] S. Mirjalili, "Genetic Algorithm," in *Evolutionary Algorithms and Neural Networks: Theory and Applications*, Cham: Springer International Publishing, 2019, pp. 43-55.

[24] C. Zhang and Y. Lu, "Study on artificial intelligence: The state of the art and future prospects," *J. Ind. Inf. Integr.*, vol. 23, p. 100224, 2021, doi:10.1016/j.jii.2021.100224.

[25] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018, doi:10.48550/arXiv.1811.03378.