International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-11 Issue-5, January 2023

Optimization Methods for Convolutional Neural Networks - The LeNet-5 Algorithm

Hamdy Amin Morsy

Abstract**: *Convolutional neural networks are an enhanced version of fully connected neural networks. The neural networks are used to recognise objects after training the neural network system on various datasets that can also be categorised into classes at the output. These networks were a breakthrough in computer vision filed for object recognition where the system can optimize its parameters for better results with using feed forward and back propagation. The convolutional neural networks reduced the time required for training and testing the dataset by replacing the full network nodes connecting to each node in the subsequent layer with some nodes or filters for each subsequent layer node. There are many algorithms for convolutional neural networks, ranging from simple algorithms to complex ones. Each algorithm has different hidden layers with distinct hyperparameters and filters. The activation functions and the number of nodes in each layer may differ for each algorithm. The applications for these convolutional neural networks span many fields, including handwritten digit recognition, handwritten alphabet recognition, and categorising groups of objects into classes, such as clothing, X-ray imaging, and many more. The LeNet-5 algorithm is a type of convolutional neural network. Through a comprehensive analysis of this algorithm, I will demonstrate that a simplified module of the algorithm can achieve maximum accuracy and a minimised loss function compared to the original algorithm.***

Keywords**: *Convolutional Neural Networks, Deep Learning, Machine Learning, Object Recognition.***

I. INTRODUCTION

Neural networks are considered to be one of the most advancing techniques in the field of computer vision in the last three decades. The neural networks have played an increasingly vital role in the design of pattern recognition, such as object recognition, which essentially focuses on digits and character recognition [1]. It can be argued that the availability of convolutional neural network (CNN) algorithms has had a significant influence on the recent success of object recognition applications. This system, in its simplest form, can be represented by a network of nodes communicating with each other to achieve the maximum performance of recognizing the input object based on training the neural network with datasets divided into classes according to the inputs [2]. The simplest way to define the concept of the neural networks is to visualize its structure as a large number of interconnected nodes organized in layers of nodes with each layer of nodes is connected to the previous and subsequent layers. The diversity and richness of natural data have been recognized from the early days of pattern recognition. A pixel-based image input can serve as the easiest form of an artificial neural network, with every pixel functioning as an input node [3], [4].

The anticipated input could be an image with features such as numbers, letters, or any other characteristic that occurs frequently and requires classification. For handwritten image classification, a 28x28 image is the most typical option. This minimum image size results in 784 pixels representing the number of nodes in the input layer. The primary structure of a simple neural network, in our case, has 784 neurons at the input layer and 10 classes at the output layer [5], [6]. Building an accurate recognition system entirely by hand is almost impossible due to the large number of interconnected neurons that exist at all layers. Each node in a layer is connected to the previous and subsequent nodes. This node has the sum of some or all of the earlier nodes, multiplying each node by a weight and adding to this sum a value called a bias.

This primary structure of a neural network is attracting considerable widespread attention from researchers due to its complexity in performing these calculations and obtaining the desired results [7], [8]. The path from the input layer to the output layer is known as the feedforward path. The weight values of each node and the bias are adaptive values that maximise the output accuracy. For the neural network system to deliver optimal performance, backpropagation is required, along with an activation function in each node.

The activation function is a method for calculating the multi-derivative of the node output function to achieve maximum accuracy in the output classification. To effectively train a neural network and achieve minimal output loss and higher entropy, a large dataset is essential. To find the derivative of a function, it has to be iteratively derivative. These functions, such as the sigmoid function, tanh function, and ReLU function, are called activation functions [9], [10].

The neural network structure shown in Fig. 1 consists of an input layer, an output layer, and multiple hidden layers that connect the input and output layers. Many proposed algorithms in neural networks are designed to estimate the real input object better. The next sections will be organised as follows: Section 2 presents the literature review on convolutional neural networks; Section 3 provides the methods and results; Section 4 and Section 5 give the discussion and conclusion. section 6.

Manuscript received on 13 November 2022 | Revised Manuscript received on 17 November 2022 | Manuscript Accepted on 15 January 2023 | Manuscript published on 30 January 2023.
* Correspondence Author
Hamdy Amin Morsy*, Assistant Professor, Department of Electronics and Communications Engineering, Faculty of Engineering, Helwan University, Cairo11792, Egypt. E-mail: hamdy_morsy@h-eng.helwan.edu.eg, ORCID ID: https://orcid.org/0000-0003-0209-0524

II. LITERATURE REVIEW

The diversity and wealth of natural data (audio, images, …) have been acknowledged from the early days of pattern recognition. The convolutional neural network (CNN) in its simplest form consists of three layers: the input layer, the output layer and one hidden layer. The complexity of CNN networks depends on the size of the input data and the number of output classes. For handwritten digit recognition, for example, the input image size is 28x28, resulting in 784 nodes at the input. On the other hand, the output classes are 10, which represent the digits from 0 to 9. In a neural network with only an input layer and an output layer, the total number of interconnected nodes, and hence the number of required weight calculations, is 7840, in addition to the 10 values for the bias. This type of structure is defined as a fully connected layer. In convolutional neural networks, there is at least one hidden layer beside the two input layers and output layer [11], [12].

The activation functions and backpropagation are primarily responsible for producing ideal weight values and biases. Equation (1) shows the matrix form of the relation between two subsequent layers, for layer h_i and layer h_j, where j=i-1, if the current layer has L nodes and the following layer has K nodes, and M rows of the weight matrix, then for a fully connected layer, $L=M$.

$$oone \begin{bmatrix} h_{i1} \\ h_{i2} \\ \cdot \\ \cdot \\ \cdot \\ h_{iK} \end{bmatrix} = \begin{bmatrix} w_{11} w_{12} & \dots w_{1L} \\ w_{21}\ w_{22} & \dots w_{2L} \\ & \dots \\ & \dots \\ & \dots \\ w_{M1} w_{M2} & \dots w_{ML} \end{bmatrix} \begin{bmatrix} h_{j1} \\ h_{j2} \\ \cdot \\ \cdot \\ \cdot \\ h_{jL} \end{bmatrix} + \begin{bmatrix} b_{i1} \\ b_{i2} \\ \cdot \\ \cdot \\ \cdot \\ b_{iK} \end{bmatrix} \quad (1)$$

For a convolutional neural network (CNN), the current layer is divided into kernels, which are small square windows of size SxS. This kernel might be called a filter, which is overlapped during the process of calculating the nodes of the next layer. The number of nodes moved through calculations is referred to as a stride, which typically has a value of 1 or 2 for most calculations.

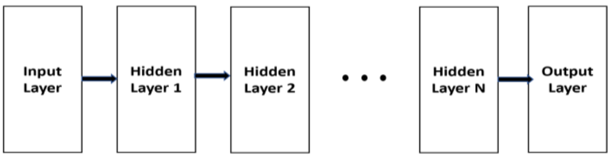

Fig. 1. Structure of a multilayer neural network

We can calculate the length of the output feature map K depending on the filter length size L, the length of the original image M and the stride s as in (2) [13], [14].

$$forK = \frac{M-L}{s} + 1 \quad for\ L < M\ , s > 0 \qquad (2)$$

Fig. 2 shows an input image of size 32x32 with one channel (grey scale image), and the next layer is 28x28 with 6 channels. The 6 channels are mathematically repetitive, but in fact, their map calculations are extracting more features from the input image [15], [16].

A kernel with size 5x5 and stride =1 is used to have the first convolutional layer in the system. For most convolutional neural networks, a second layer is added to the system to extract more essential features of the input layer. This process is known as pooling, and there are two common types of pooling: maximum pooling and average pooling. In maximum pooling, a 2x2 window is mapped over the input layer with a stride, resulting in halving the input layer (from 14x14 to 7x7). Several factors influence the number of convolutional layers and the number of fully connected layers. For large datasets, you might need many hidden layers for training and testing.

There are two key terms—feedforward and backpropagation—that describe the transmission of data in neural networks. The feedforward transmits the features from the input to the output. In backpropagation, the information is fed back to optimize the weights and biases. For that purpose, an activation function is crucial for achieving this objective. An activation function with a multi-derivative property is mandatory to achieve the best results. Activation functions, then, are used to provide the neural network some non-linearity. The most common activation functions with multi-derivative properties are the sigmoid function, the Tanh function and the leaky ReLU function. Equations (3) through (5) stated these different activation functions [17], [18].

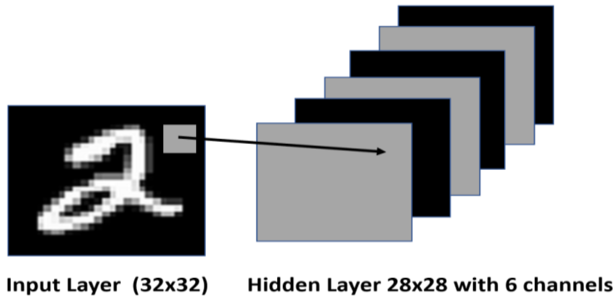

Fig. 2. The first Convolution Layer with 6 Channels

$$f(x) = \frac{1}{1+e^{-x}} \qquad (3)$$

$$f(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})} \qquad (4)$$

$$f(x) = \max(0.1x, x) \qquad (5)$$

Fig. 3, Fig. 4 and Fig. 5 show the activation functions for the sigmoid function, the tanh function, and the leaky ReLU function and their derivatives, respectively. The last layer which represents the output class is utilizing a softmax function as shown in (6) which provide outputs that summed up to one.

$$.(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \qquad (6)$$

There are many techniques for CNN network structures, starting with convolutional network layers and pooling layer pairs, which may be repeated once or twice in the structure, using max pooling or average pooling, and ending with fully connected layers, followed by the output layer, which provides the classes of the input [19], [20]. One of the most introductory methodologies in convolutional neural networks is the LeNet-5 algorithm [1]. This algorithm was introduced in 1998 by Lecun and his colleagues as a new technique for pattern recognition. The LeNet5 algorithm has 5 layers with an input layer of 32x32 nodes representing the size of the input image. The type of dataset used for training and

Retrieval Number: 100.1/ijrte.E73550111523
DOI: 10.35940/ijrte.E7355.0111523
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

testing is handwritten digits, which have 10 classes at the output.

A regularization function is used to limit the raining error. Regulating the trade-off between decreasing the training error and minimizing the predicted gap between the training error and test error is done by determining the capacity of the accessible subset of the parameter space [17].

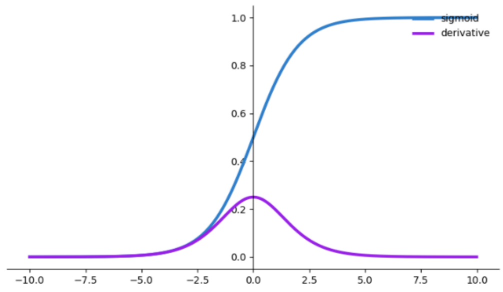

Fig. 3. Sigmoid function and its derivative

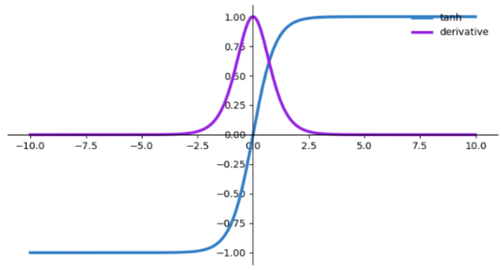

Fig. 4. Tanh function and its derivative

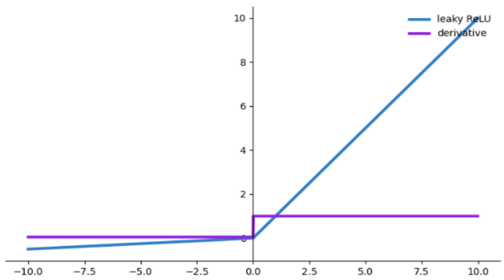

Fig. 5. Leaky ReLU function and its derivative

Assume the convolutional layer symbol is C and the sub-sampling layer symbol is S. The C1 layer has 6 feature maps and a size of 28x28, resulting from using a filter with size 5x5 and a stride of one. The first layer has 28x28x26x6 = 122304, resulting from 6 channels times 28x28 times 26, which is the size of the filter plus the bias. The second layer, S2, is defined as a subsampling layer with a filter size of 2x2 and a stride of 2. The size of this layer is six channels, each with a size of 14 × 14, resulting in 5,880 connections.

The third layer, C3, is a convolutional layer with 16 feature maps of size 10 × 10 and a 5 × 5 filter with a stride of 1. This structure of layer C3 results in 41600 connections. The Layer S4 is another subsampling layer with a size of 16x5x5 and a total of 2,000 connections. The fifth layer, C5, is a convolutional layer with 120 feature maps and a total of 48,120 connections. The sixth layer, F6, is a fully connected layer with 10,164 connections and trainable parameters. The final layer is 10 10-class output layer. The LeNet-5 algorithm was based on learning rates of around 0.0001, with a training error rate of 0.3% and a test error rate of approximately 1%. This algorithm was one of the earliest convolutional neural network algorithms, marking the beginning of the development of deep learning.

III. METHODS

The technique used here involves altering certain LeNet-5 algorithm functions and comparing the performance of these new systems with that of the original convolutional network algorithm. The results will be evaluated for the effectiveness of various learning methods on a dataset of handwritten digit recognition. The feature maps of each layer will be changed to find the maximum output accuracy for each feature map change in each layer. The rate of change of output accuracy with respect to feature maps will be calculated to find the optimum feature map values for the best output accuracy. Additionally, some of the most common activation functions will be utilised in this system, including sigmoid, tanh, ReLU, and leaky ReLU. The last two layers will be removed interchangeably to measure the output accuracy.

IV. RESULTS

As shown in Fig. 6, the feature maps in layer C1 are changed from 4 to 8, and the feature maps in layer C3 are changed from 14 to 18 to find the maximum output accuracy that results from these values. The hyperparameters used in this system are learning rate = 0.001, batch size = 64, and number of epochs = 10. The activation function used is the ReLU activation function. The highest accuracy was reached in this system by applying 5 feature maps at layer C1 and 18 feature maps at layer C3.

I repeated the training and testing for the proposed system with layer C1, with five feature maps, and layer C3, with 18 feature maps. The maximum accuracy achieved with these hyperparameters is 99.15%. As shown in Table II operated the system with different activation functions and pooling methods. The maximum accuracy was achieved with the sigmoid activation function and average pooling, using 5 and 18 feature maps at layers C1 and C3, respectively. I tested the system for different batch sizes (64, 100, and 200), and the output accuracy was 99.24%, 99.12%, and 99.24%, respectively.

I tested the system for removing one of the last two fully connected layers, using different batch sizes with the C5 layer fully connected and the F6 layer removed. For batch sizes of 64 and 32, the outputs are 99.21% and 99.17%, respectively. By utilising the F6 layer and removing the C5 layer, the output results for 64 and 32 batch sizes are 99.09% and 99.24%.

V. DISCUSSION

The tested neural network system has many hyperparameter changes and techniques. From the number of layers to batch size and number of epochs. The results of this type of system fluctuate around a specific value, which is a converging function for the normal and expected behaviour of the system.

Retrieval Number: 100.1/ijrte.E73550111523
DOI: 10.35940/ijrte.E7355.0111523
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The processing time of training datasets affects the output results for large datasets.

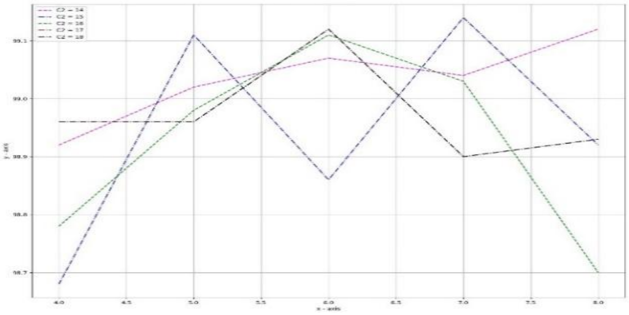

Fig. 6. Leaky ReLU function and its derivative

Table I. Result for different activation functions and pooling.

	Sigmoid	Tanh	ReLU	Leaky ReLU
Max Pooling	98.86%	98.78%	99.06%	99.13%
Average Pooling	99.14	98.93	99.08%	99.1

Removing one fully connected layer from the system doesn't affect the output results, even removing the larger nodes layer (120 nodes) improves the output result. The type of activation function affects the output results. The sigmoid activation function with average pooling provides higher performance than other functions. The results of utilising a batch size of 32 provide good outputs, as a small batch size will help avoid the problem of overfitting.

VI. CONCLUSION

Convolutional neural networks offer various system structures to achieve optimal performance for training and testing datasets. The LeNet-5 algorithm was the cornerstone of new CNN techniques. The proposed system proved to have better performance than the original system. With one layer less in the CNN structure, the output has higher performance than the original system. The CNN system should be tested for different activation functions and pooling functions. In this proposed system, the sigmoid activation function provides higher accuracies for output classifications. The changes in batch size also affect the output results; for smaller batch sizes, higher performance is obtained.

DECLARATION

Declaration	If not applicable
Funding	Not funding.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethics Approval	Not relevant
Consent to Participate	Not applicable
Consent for Publication	Not applicable
Availability of Data and Materials	Not applicable
Authors contributions	Not applicable
Code Availability	Not applicable

REFERENCES

1. Y. Lecun, L. Bottou, Y. Bengio, and P. Ha, "Gradient-Based Learning Applied to Document," Proc. IEEE, no. November, pp. 1–46, 1998, doi: 10.1109/5.726791. [CrossRef]
2. M. Nielsen, Neural Networks and Deep Learning. 2018. doi: 10.1201/b22400-15. [CrossRef]
3. M. Ramzan et al., "A survey on using neural network-based algorithms for handwritten digit recognition," Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 9, pp. 519–528, 2018, doi: 10.14569/ijacsa. 2018.090965. [CrossRef]
4. O. I. Abiodun et al., "Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition," IEEE Access, vol. 7, no. February 2017, pp. 158820–158846, 2019, doi: 10.1109/ACCESS.2019.2945545. [CrossRef]
5. H. A. Morsy, "Performance Analyses of the Eastern Arabic Hand Written Digits Recognition Using Deep Learning," Am. J. Sci. Eng. Technol., vol. 7, no. 3, pp. 57–61, 2022, doi: 10.11648/j.ajset.20220703.11.
6. K. T. Islam, G. Mujtaba, R. G. Raj, and H. F. Nweke, "Handwritten digits recognition with artificial neural network," in 2017 International Conference on Engineering Technology and Technopreneurship (ICE2T), Sep. 2017, pp. 1–4. doi: 10.1109/ICE2T.2017.8215993. [CrossRef]
7. H. A. Morsy, "Developing a New CNN Technique for Arabic Handwritten Digits Recognition," Int. J. Recent Technol. Eng., vol. 9, no. 3, pp. 520–524, 2020, doi: 10.35940/ijrte.c4588.099320. [CrossRef]
8. J. H. Alkhateeb and M. Alseid, "DBN-Based learning for Arabic handwritten digit recognition using DCT features," 2014 6th Int. Conf. Comput. Sci. Inf. Technol. CSIT 2014 - Proc., no. September, pp. 222–226, 2014, doi: 10.1109/CSIT.2014.6806004. [CrossRef]
9. H. A. Morsy, "Optimization of Arabic Handwritten digits recognition using CNN," Int. J. Sci. Eng. Res. V, vol. 11, no. 11, pp. 372–376, 2020.
10. D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015.
11. C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," in 2nd International Conference on Computational Sciences and Technologies, 17-19 Dec 2020 (INCCST 20), Dec. 2020, pp. 124–133.
12. J.-C. Vialatte, V. Gripon, and G. Coppin, "Learning Local Receptive Fields and their Weight Sharing Scheme on Graphs," Jun. 2017. [CrossRef]
13. S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," Proc. 2017 Int. Conf. Eng. Technol. ICET 2017, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/ICEngTechnol.2017.8308186. [CrossRef]
14. V. Shetty, M. Singh, S. Salunkhe, and N. Rathod, "Comparative Analysis of Different Classification Techniques," SN Comput. Sci., vol. 3, no. 1, p. 50, Jan. 2022, doi: 10.1007/s42979-021-00906-z. [CrossRef]
15. V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," pp. 1–31, 2016, [Online]. Available: http://arxiv.org/abs/1603.07285
16. K. Sanghvi, A. Aralkar, S. Sanghvi, and I. Saha, "A Survey on Image Classification Techniques," SSRN Electron. J., 2020, doi: 10.2139/ssrn.3754116. [CrossRef]
17. C. F. G. Dos Santos and J. P. Papa, "Avoiding Overfitting: A Survey on Regularization Methods for Convolutional Neural Networks," ACM Comput. Surv., vol.. 54, no. 10s, pp. 1–25, 2022, doi: 10.1145/3510413. [CrossRef]
18. T. DeVries and G. W. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout," 2017, [Online]. Available: http://arxiv.org/abs/1708.04552
19. T. Wiatowski and H. Bolcskei, "A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction," IEEE Trans. Inf. Theory, vol. 64, no. 3, pp. 1845–1866, 2018, doi: 10.1109/TIT.2017.2776228. [CrossRef]
20. H. Leterme, K. Polisano, V. Perrier, and K. Alahari, "On the Shift Invariance of Max Pooling Feature Maps in Convolutional Neural Networks," pp. 1–17, 2022, [Online]. Available: http://arxiv.org/abs/2209.11740

AUTHORS PROFILE

Hamdy Amin Morsy (MSc 2002, PhD 2012) is an assistant professor at Helwan University, Faculty of Engineering, Electronics and Telecommunications Engineering Department, Cairo, Egypt. He is also a visiting professor at Qassim University in the Kingdom of Saudi Arabia. He received his Master's degree from Stevens Institute of Technology in Hoboken, NJ, USA, in 2002, and his PhD from Helwan University in Cairo, Egypt.