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 enhanced version of fully connected neural networks. The neural networks are used to recognize objects after training the neural network system for some datasets that can also be divided 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 of training and testing the dataset by replacing the full network nodes connecting to each node in the subsequent layer to some nodes or filter to 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 different hyper parameters and filters. The activation functions and number of nodes in each layer for each algorithm may be different. The applications for these convolutional neural networks cover many fields such as hand written digit recognition, alphabet handwritten recognition, and any group of objects that can be divided into classes such as cloth, X-ray imaging and many more. The LeNet-5 algorithm is one of the convolutional neural networks. With full analysis of this algorithm, I will prove that a simple module of the algorithm can provide maximum accuracy and minimum loss function than 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. As a matter of fact, the neural networks have played an increasingly vital role in the design of pattern recognition such as object recognition which essentially focused on digits and character recognition [1]. In fact, it could be argued that the availability of convolutional neural networks (CNN) algorithms has been an imperative influence in the recent success of object recognition applications. This system in its simplest form can represented by a network of nodes communicated together to highly 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 like numbers, letters, or any other feature that could have an excessive number of occasions and demand for 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 owing to the large number of interconnected neurons existing 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 previous nodes multiplying each node with a weight and adding to this sum a value called a bias.

This primary structure of a neural network is attracting considerable widespread attention of the researchers due to its complexity of performing these calculations and obtaining the desired results [7], [8]. The path from the input layer to the output layer is the feed forward path. The weight value of each node and the bias are adaptive values to maximize the output accuracies. For the neural network system to deliver the best performance, back propagation is required with activation function in each node.

The activation function is a method for multi derivative of the node output function to achieve maximum accuracy of the output classification. In order to effectively train a neural network and ultimately provide minimum 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 sigmoid function, tanh function, and ReLU function are called activation functions [9], [10].

The neural network structure shown in Fig. 1 has input layer, output layer and multi hidden layers connected in between input and output layers. There are many proposed algorithms in neural networks are designed to better expected the real input object. The next sections will be organized as follows: Section 2 will present the literature review on convolutional neural networks; Section 3 provides the methods and results in section 4, section 5 provide the discussion and conclusion in 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 input data size and output classes. For handwritten digits recognition as an example the input image size 28x28 which results in 784 nodes at the input. On the other hand, the output classes are 10 which represents the digits from 0 to 9. In a neural network with only input layer and 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 convolution neural networks, there is at least one hidden layer beside the two input layer and output layer [11], [12].

The activation functions and back-propagation are primarily responsible for the capability to produce 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 subsequent layer has K nodes, and M rows of weight matrix, then for a fully connected layer $L=M$.

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

For convolutional neural network (CNN), the current layer is divided into kernels which is a small square window *SxS* matrix. 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 called a stride which has a value of 1 or 2 for most calculations.

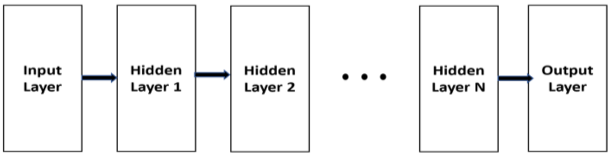

Fig. 1. Structure of multilayer neural network

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].

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

Fig. 2 shows an input image with 32x32 size with one channel (gray scale image) and the next layer is 28x28 with 6 channels. The 6 channels are in mathematical way repetitive channels but in feature 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 much important features of the input layer. This process is called pooling, there are two common types of pooling: maximum pooling and average pooling. In maximum pooling, a window of 2x2 is mapped over the input layer with stride=2, this will result in halved the input layer (14x14). There many factors affecting the number of convolutional layers and the number of fully connected layers, for large size 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 multi derivative property is mandatory to accomplish 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 sigmoid function, Tanh function and leaky ReLU function. Equations (3) through (5) stated these different activations functions [17], [18].

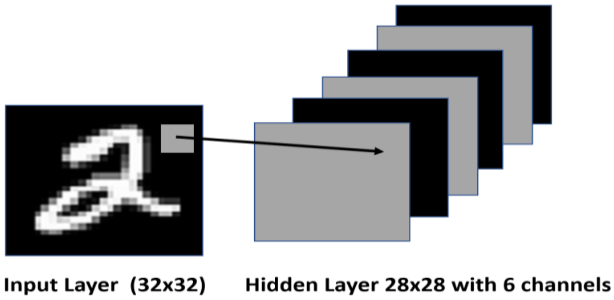

Fig. 2. The first Convolution Layer with 6 Channels

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

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

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

Fig. 3, Fig. 4 and Fig. 5 show the activation functions for sigmoid function, tanh function, and 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.

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

There are many techniques for CNN network structure starting using convolutional network layer and pooling layer pairs which might repeated once or twice in the structure using max pooling or average pooling and ending with fully connected layers then the output layer which provide the classes of the input [19], [20]. One of the most introductory methodology in convolutional neural networks is the LeNet-5 algorithm [1]. This algorithm appeared in 1998 by Lecun and his colleagues for the approach of 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 trained and tested is handwritten digits which has 10 classes at the output.

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)

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

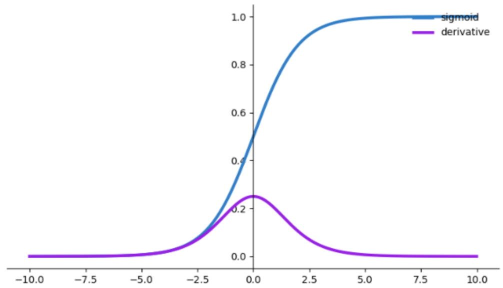

Fig. 3. Sigmoid function and its derivative

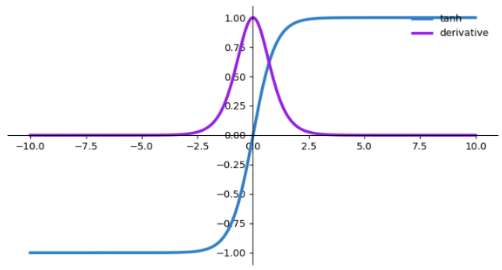

Fig. 4. Tanh function and its derivative

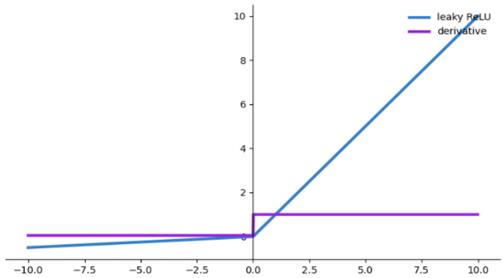

Fig. 5. Leaky Re LU function and its derivative

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 28x28 resulting from using a filter with size 5x5 and stride equals to 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 is S2 which is defined as a sub-sampling layer with a filter size of 2x2 and a stride equals to 2. The size of this layer is 6 channels with size 14x14 which results in 5,880 connections.

The third layer C3 is a convolutional layer with 16 feature maps of size 10x10 and a filter of size 5x5 with stride equals to one. This structure of layer C3 results in 41600 connections. The Layer S4 is another sub-sampling layer with a size of 16x5x5 and total connections of 2,000. The fifth layer C5 is a convolutional layer of 120 feature maps and total connections of 48,120. The sixth layer F6 is a fully connected layer with 10,164 connections and trainable parameters. The final layer is 10 classes output layer. The LeNet-5 algorithm was based on learning rates around 0.0001 with train error rate equals 0.3% and test error rate around 1%. This algorithm was one of the earliest convolutional neural network algorithms which established the development of deep learning.

III. METHODS

The technique used here involves altering certain LeNet-5 algorithm functions and contrast 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 data sets 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 maps values for the best output accuracy. Also, some of the most common activation functions will be utilized in this system such as sigmoid, tanh, ReLU, and leaky ReLU. The last two layer 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 is changed from 14 to 18 to find the maximum output accuracy that result from these values. The hyper parameter used in this system are learning rate=0.001, batch size = 64 and number of epochs = 10. The activation function used is ReLU activation function. The highest accuracy 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 5 feature maps and layer C3 with 18 feature maps. The maximum accuracy achieved with these hyper parameters is 99.15%. As shown in Table I, the I operated the system with different activation functions and different pooling. The maximum accuracy obtained with sigmoid activation function and average pooling with 5 and 18 feature maps at layers C1 and C3 respectively. I tested the system for different batch size 64, 100, 200, the output accuracy was 99.24%, 99.12% and 99.24% respective.

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

V. DISCUSSION

The tested neural network system has many hyper parameters changes and techniques. From number of layers to batch size and number of epochs. The results of this type of system are fluctuating around a certain value which is a converging function for the normal and expected right behavior of the system.

Retrieval Number: 100.1/ijrte.E73550111523
DOI: [10.35940/ijrte.E7355.0111523](https://doi.org/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.

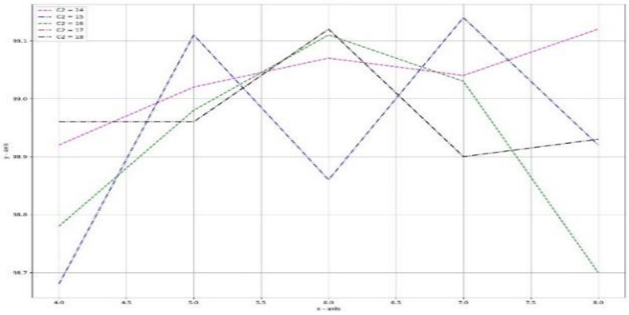

Fig. 6. Leaky Re LU function and its derivative

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 utilizing a batch size of 32 provides good outputs since small batch size will avoid the problem of overfitting.

VI. CONCLUSION

The Convolutional neural networks provide varieties of system structure to achieve the best performance of training and testing dataset. The LeNet-5 algorithm was the corner stone 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, sigmoid activation function provides higher accuracies for output classifications. The changes in batch size also affects the output results, for smaller batch size, higher performance is obtained.

DECLARATION

Declaration	If not applicable
Funding	Not funding.
Conflicts of Interest/ Competing Interests	Not 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 Material	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 hand written 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, Saudi Arabia Kingdom. He received his Master degree from Stevens Institute of Technology, Hoboken, NJ, USA in 2002, and his PhD from Helwan University, Cairo, Egypt

.

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