International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-3 September 2020

Developing a New CNN Technique for Arabic Handwritten Digits Recognition

Hamdy Amin Morsy

Abstract: Convolutional Neural Networks (CNN) have many applications in object recognition such as character and digit recognition. Few researches are performed on Arabic handwritten digits recognition. In this research, we will develop a new algorithm to utilize the convolutional neural networks with sigmoid function (σ-CNN) to recognize Arabic handwritten digits recognition. The performance of this method provides minimum cost functions with maximum testing accuracy results in compared to other existing techniques.

Keywords: Machine Learning, Neural Networks, Image Processing, Natural Language Processing

I. INTRODUCTION

Neural networks filed is one of the very promising fields in the area of multimedia detection including texts, sounds and images. One of the most important fields is the handwritten recognition. The neural networks principle is based on measuring some features of the inputs to compare with expected output. To do that many input data are need to have complete picture of the input. In handwritten recognition, the input data is images of the characters in our case Arabic digits. The neural network implementation needs many input images maybe hundreds or thousands to have high accuracy of the digits recognition. These images entered to the system is called training images or training dataset [1]. There are many applications to neural networks such as flight control, robotics and medical systems recognition [2]. The neural network training is based on stochastic gradient descent algorithm with utilization of sigmoid function.

The neural networks from the name implied is behaving like a human brain, which contains neurons and links to these neurons. Each neuron has a bias and many links to that neuron. The links connects different neurons with different functions; it connects between inputs and outputs through many neurons, which we call hidden layers. Each link has a weight which, proportion to the importance of the information being transferred. The neural network is basically has some neuron neurons maybe tens or hundreds and maybe thousands to process the input information. The information transferred from one neuron in a layer to another neuron in next layer is processed in the new neuron and so on.

Manuscript received on 01 August 2020 | Revised Manuscript received on 05 August 2020 | Manuscript Accepted on 15 September 2020 | Manuscript published on 30 September 2020.

Correspondence Author

Hamdy Amin Morsy,* Department of Electronics and Communications Engineering, Faculty of Engineering at Helwan University, Cairo, Egypt

Email: hamdy_morsy@h-eng.helwan.edu.eg

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

Each link between two neurons has a weight and each neuron has a bias. These two variables are the cornerstone for training the neural network. The training system is based on changing the values of weights and biases to achieve maximum matching between the input and the output with minimum errors during the training time

There are many research papers focused on English handwritten digits and a few considered Arabic handwritten recognition. In this paper, our focal point will be on the Arabic handwritten digits recognition. The Arabic digits consist of 10 different digits ( )the zero matches the full stop in English and the two digit is slightly different from the digit three which makes the training programs is a little bit harder to get a good result. Each neuron or perceptron will have a weight from other neurons and a bias to process the feedforward propagation process. Adjusting the values of weights and biases for all neurons and the number of layer will help achieve perfect matching. For a computer program to process new data – in this case Arabic handwritten digits- training dataset will be collected so the computer program can learn the data to be studied and a validation data and test data are need to check for the program performance to manage and handle the input data. The computer model received first a training dataset may be hundreds and thousands of images containing the required information to be trained for and a test dataset to measure the model performance [3]. Some images may need image resized or image scaling to achieve good results with neural networks applications [4,5]. The rest of the paper will be organized as follows: section II introduces the neural networks and sigmoid function. Section III provides the comparisons and results. Conclusion will be provided in section IV.

II. NEURAL NETWORKS AND SIGMA FUNCTION

The simplified form of a convolutional neural network CNN consists of three layers. These layers are input layer, output layer and hidden layer. For handwritten digits recognition whether it is Arabic or English, the input layer has number of neurons corresponds to the number of pixels of the input image which contains the digit. In our case a 28x28 image pixels standard will be used as a size for training and testing data [6]. Figure 1 shows the CNN network three layers, one input layer which has 784 neurons, one output layer which has 10 neurons and the hidden layer which contains a number of neurons between 10 and 784. The number of hidden layer neurons is some number between the geometric mean and the square root of the input layer, which in our case is between 90 and 30 neurons to achieve good results. Each perceptron takes some factor which affects its value, the factors are bias and weights from previous layer's perceptrons.

Any change in the values of weights and bias may cause the behavior of the rest of the network to completely change in some very complicated way. A perceptron takes several binary inputs, $x_1, x_2, \ldots, x_n$ and produces a single binary output: Rosenblatt proposed a simple rule to compute the output as a function of the input or previous layer perceptrons. Using these calculations, there is a threshold to determine the perceptron output whether it is 0 or one. [7-9].

As shown in Fig. 1 each neuron has multiple inputs and multiple outputs from different neurons, which affect its activation value.

$$output = \begin{cases} 0 \;\; if\; w.x + b \le 0 \\ 1\; if\; w.x + b > 0 \end{cases} \qquad (1)$$

In equation (1), the weights w and the input x are formulated as vectors, the value b is the neuron bias and the output is represented as 0 or 1. While a perceptron has an output of a 0 or 1 depending on the input weights and bias, the sigmoid neuron has an output ranging from 0 and 1 for example 0.6 , 0.9 and so on[10-11]. The output in this case, takes the value of $\sigma(w.x+b)$, where σ is called the sigmoid function, and is defined by[1]:

$$\sigma \equiv \frac{1}{1+e^{-z}} \qquad (2)$$

The complete neuron equation as a function of weights and bias is given as:

$$\sigma = \frac{1}{1+exp(-\sum_j w_j x_j - b)} \qquad (3)$$

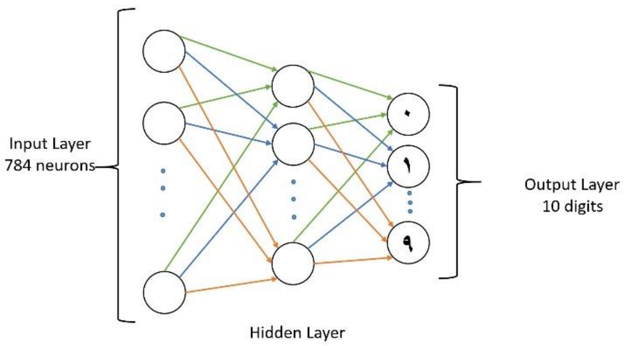

Fig. 1 A simplified CNN network

The σ function is the smooth curve of the step function, which produces output 0 and 1 as in perceptron. By using the actual σ function, we get a smoothed out perceptron [12-13].

The neural network shown in Fig. 1 has three layer, the first layer which has the input neurons, the second layer which is called hidden layer and finally the third layer which has the output neurons [14-15]. This is the simplest neural network structure, some networks can have many hidden layers and can reach up to hundreds of layers.

The output layer of the network shown in Fig. 1 has 10 neurons. Each neuron corresponds to one desired output in our case, these neurons represents the Arabic digits from (0) to (9). For each input, the output layer makes a decision depending on the value of each output neuron. For example if the first neuron has a maximum out, then the output is the Arabic digit (0) and so on. The value of each neuron is called the activation value [16-17].

The accuracy of the neural network is mainly depending on the number of training dataset images. We have collected handwritten Arabic digits from students and using the data augmentation to increase the dataset size. The image dataset is divided into three groups the first group is the training images and has 50,000 images, the second group is the validation images which has 10,000 images the last group is the test images which has 10, 000 images. Each images is a two dimensional grayscale image with 28x28 size [18].

Since the image size is 28x28 pixels, the input layer will have 784 neurons with each neuron i has input x_i. The total input can be represented with a vector x containing 784 elements. The output layer on the other hand has 10 outputs representing the Arabic 10 digits y = y(x), with y having 10 dimensional vector. The next step is to find a technique to find the values of weights and biases for all neurons so we can minimize the accuracy errors. Let's define the cost of weight w and bias b as:

$$C(w,b) \equiv \frac{1}{2n}\sum_x \|y(x) - a\|^2 \qquad (4)$$

Equation (4) can be called the quadratic cost function It describes the mean square difference between the output a and the expected output y(x). The C(w, b) is zero when the output equals to the expected output. In this case, the neural network has accuracy of 100% and 0% error. Now, our goal is to find value for weights an biases to reach maximum accuracy with minimum error. Stochastic gradient descent SGD algorithm can be utilized to achieve this goal [19].

Writing out the gradient descent update rule in terms of components, we have

$$w_k \rightarrow \quad w'_k = w_k - \eta \frac{\partial C}{\partial w_k}$$

$$b_l \rightarrow \quad b'_l = b_l - \eta \frac{\partial C}{\partial b_l} \qquad (5)$$

Equation (5) describes the relation between weights and biases with the partial derivative of both weight and bias with learning rate η. Having many training inputs, a minimum of w and b can be achieved which in turn minimize the total cost of the output. We divide the training data to batches of small batches around 10 inputs. The input batch to the neural network is called an epoch of training [20].

III. COMPARISON AND RESULTS

The Arabic handwritten digits consists of 10 digits. The training dataset has 60,000 training images and 10,000 test images.

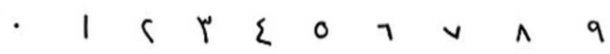

Fig. 2 An example of Arabic handwritten digits

Fig. 2 shows an Arabic handwritten digits which will be used for training and testing data.

1- Input: training, testing data, labels

2- Process: initialize random weights and biases using Numpy library

3- Start stochastic gradient descent SGD algorithm

4- Repeat the program

5- If the cost function is minimum

6- Exit

7- Output: test data with minimum cost

Fig. 3 The Algorithm

Fig. 3, shows the algorithm proposed to Arabic handwritten digits recognition. The algorithm is written in Python language and using Numpy library for doing fast linear algebra. The weights and biases are initialized randomly using Gaussian distributions with mean 0 and standard deviation 1.

The output activations vector of the third layer of neurons is given by:

$$a' = \sigma(wa + b) \qquad (6)$$

Each image in the training dataset has two values the training input which is the input digit x and the desired output y. The proposed algorithm will measure the weight w and bias b and also calculates the accuracy and error after each epoch.

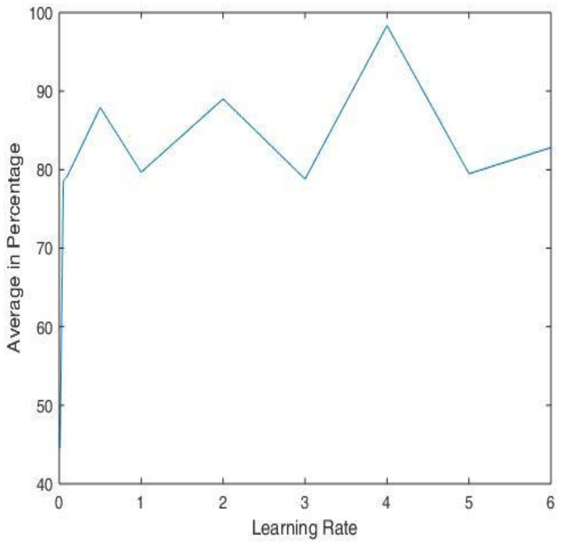

Fig. 4 The average Percentage vs Learning rate for 60 neurons hidden layer

After calculating the weight and biases, the backpropagation is utilized to minimize the cos function C at the output. We will apply the algorithm to a number of epochs 30 with a batch size of 10 and a learning rate of η = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}x^{-3} and the number of hidden layer neurons is 60 to find the minimum cost function. Fig. 4, shows the average percentage of output test images against learning rate. The maximum three values of average percentages are at learning rate 0.008, 0.006, and 0.004 respectively. Fig. 5 shows the cost function against epochs for the three maximum values. At learning of η=0.008 a minimum cost function is obtained.

Fig. 6 shows the average percentage of the corrected output against number of neurons. The average percentage has maximum average percentage at neurons equals 30, 60, 50 respectively. It is shown in the Fig. that with 30 neurons of the hidden layer, we obtain a maximum average percentage. Fig. 7 shows the detail of these neurons with maximum average percentage. With 60 neurons, the cost function is minimum. We also run the program for 10 time for values of 60 neurons and learning rate η=0.008 as shown in Fig. 8. This algorithm provides good results with minimum number of iterations.

Table 1 shows the cost function for different techniques of Arabic handwritten digits recognition. Takruri et al. [21] presented three level classifiers based on SVM Machine, Fuzzy C Means. They tested the new algorithm on a public dataset. The dataset contains 3510 images with 40% are used for testing and 60% of images are used for training. The overall cost functions reported is 12%. AlKhateeb et al. [22] presented a technique to classify Arabic handwritten digit recognition using Dynamic Bayesian Network DBN. They used DCT features for classification. They used ADBase dataset for trained and tested on Arabic digits database. Their technique provides cost function of 14.74%. Majdi Salameh [23] presented two techniques for enhancing recognition rate for typewritten Arabic digits. These proposed techniques are implemented and tested on some fonts. Their methods provid 5% cost function. Melhaoui etal.[24] presented Arabic numerals recognition based on an improved version of the loci characteristic. They applied their technique on 400 training digits and 200 testing images. They calculated the cost function for the training digits which was 1%. Pandi Selvi and Meyyappan [25] presented a technique to recognize Arabic digits using back propagation neural networks. Their technique shows that the proposed method provides 4% cost function for a small sample handwritten database. Mahmoud [26] presented Arabic (Indian) handwritten digits recognition using Gabor-based features. He used 21120 images with 70% training digits and 30% testing images. He achieved good results with 0.15% and 2.16% cost function for training and testing digits respectively. El-Sawy et al. presented CNN for Handwritten Arabic digits recognition based on LeNet-5. They used MAD Base dataset for 60,000 training digits and 10,000 testing digits. They achieved 1% and 12% for training and testing digits cost function respectively. Our approach proved superior outcomes with 0.92% for testing digits which outperform the existing techniques for Arabic handwritten recognition.

IV. CONCLUSIONS

The convolutional neural networks (CNN) and sigmoid function are applied to recognize Arabic handwritten digits recognition. The algorithm applied run many times to achieve minimum error rates with maximum accuracy. The results achieved using this method is very promising in handwritten recognition. In addition, the number of epochs and the learning rate can be optimized using our algorithm to achieve minimum cost function.

Table I Comparisons of different techniques vs testing accuracy

Authors	Database	Train, Test	Cost %
Takruri [21]	Public	2106, 1404	12
AlKhateeb [22]	ADBase	60,000, 10,000	15
Salameh [23]	Fonts	1000, 1000	5
Melhaoui [24]	Private	600, 400	1
Selvi [25]	Private	Unknown	4
Mahmoud [26]	Private	14784, 6336	0.15, 2.16
El-Sawy [27]	MADBase	60,000, 10,000	1, 12
Proposed	Private	50,000, 10,000	0.92

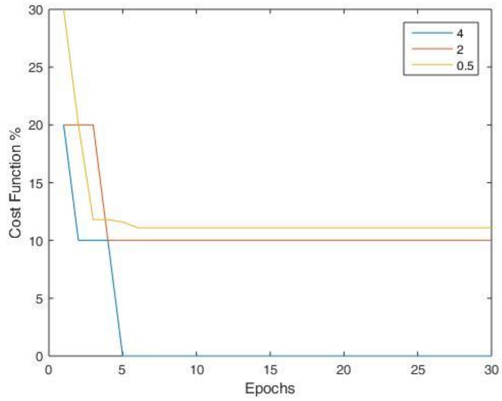

Fig. 5 The Cost function vs Epochs for learning rate (8, 6, 4)x^{-3} and Neurons=60

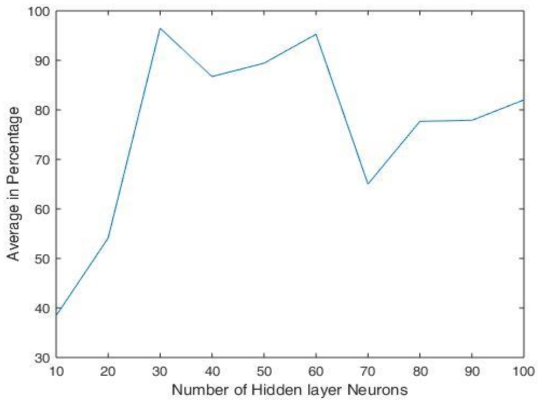

Fig. 6 The Number of Hidden layer Neurons vs Average in Percentage

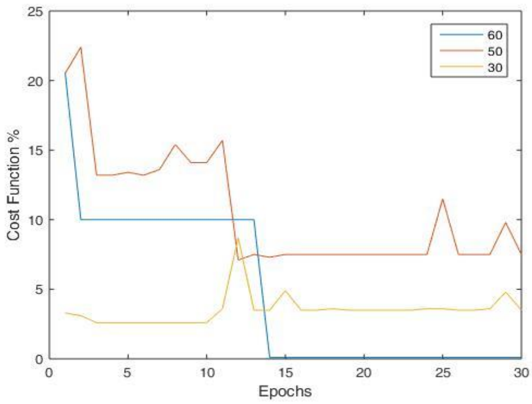

Fig. 7 Number of Epochs vs Cost Function for Neurons (60, 40, 30)

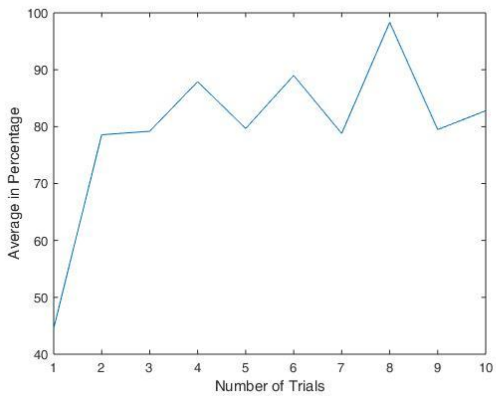

Fig. 8 Number of Trials vs the average in percentage for Neurons =60, and 0.008

REFERENCES

1. A.Charu, "Neural Networks and Deep Learning: A Textbook", Springer, 1st ed. 2018 Edition.
2. Oludare I. Abiodun et al, "Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition", DOI 10.1109/ACCESS.2019.2945545, IEEE Access.
3. S. Al-Ma'adeed, D. Elliman, C. A. Higgins, "A data base for Arabic handwritten text recognition research", In Proceedings eighth international workshop on frontiers in handwriting recognition, IEEE 2002, pp 485–489.
4. H. A. Morsy, "Comparison of Commonly Used Non-Adaptive Image Scaling Techniques",CiiT Digital Image Processing 10 (9), 177-180.
5. H. A. Morsy, H, "Performance Analysis of the Effects of Non-Adaptive Image Scaling on Image Edges", IJRTE 7 (6), 1692 – 1696.
6. D. C. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber, "Convolutional neural network committees for handwritten character classification", In 2011 International conference on document analysis and recognition, pp 1135–1139. IEEE. [CrossRef]
7. A. El-Sawy, M. Loey, E. Hazem, "Convolutional neural network for Arabic handwritten characters dataset classification" , https://www.kaggle.com/mloey1/cnn-for-arabic-handwritten-characters/comments. Accessed: 30 May 2019.
8. I. Goodfellow, Y. Bengio, A. Courville,"Deep learning. MIT Press", 2016,http://www.deeplearningbook.org.
9. S. Impedovo, F. M. Mangini, D. Barbuzzi, "A novel prototype generation technique for handwriting digit recognition", Pattern Recognit, 2014, 47(3):1002–1010. [CrossRef]
10. G. Latif, J. Alghazo, L. Alzubaidi, M. M. Naseer, Y. Alghazo, "Deep convolutional neural network for recognition of unified multi-language handwritten numerals" In 2018 IEEE 2nd International workshop on Arabic and derived script analysis and recognition (ASAR), pp 90–95. [CrossRef]
11. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al, "Gradient based learning applied to document recognition", Proc IEEE, 1998, 86 (11):2278–2324. [CrossRef]
12. A. Mars, G. Antoniadis, "Arabic online handwriting recognition using neural network", Int J ArtifIntell Appl(IJAIA), 2016,7(5). [CrossRef]
13. M. A. Mudhsh, Almodfer R, "Arabic handwritten alphanumeric character recognition using very deep neural network", 2017, Information 8(3). [CrossRef]
14. M. Ramzan, H. U. Khan, S. M. Awan, W. Akhtar, M. Ilyas, A. Mahmood, A. Zamir, "A survey on using neural network based algorithms for hand written digit recognition", 2018, Int J Adv Comput Sci Appl. https://doi.org/10.14569/IJACSA.2018.090965. [CrossRef]
15. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", 2005, J Mach Learn Res 15(1):1929–1958.
16. J. Wu, "Introduction to convolutional neural networks. National Key Lab for Novel Software Technology", 2017, vol 5, Nanjing University, China, p 23.
17. K. Younis, "Arabic handwritten characters recognition based on deep convolutional neural networks", 2018, Jordan J Comput Inform Technol (JJCIT) 3. [CrossRef]
18. U. R. Babu, Y. Venkateswarlu, A. K. Chintha, "Handwritten digit recognition using k-nearest neighbor classifier", In Proceedings of the 2014 World Congress on Computing and Communication Technologies, (WCCCT 2014), Trichirappalli, India, 27 February–1 March 2014; pp. 60–65.
19. X. X. Niu, C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits", Pattern Recognit. 2012, 45, 1318–1325. [CrossRef].
20. F. A. Al-omari, O. Al-jarrah, "Handwritten Indian numerals recognition system using probabilistic neural networks", Adv. Eng. Inform. 2004, 18, 9–16. [CrossRef]
21. M. Takruri, R. Al-Hmouz, A. Al-Hmouz, "A three-level classifier: fuzzy C means, support vector machine and unique pixels for Arabic handwritten digits", In: World Symposium on Proceedings of Computer Applications & Research (WSCAR), pp. 1—5 (2014). [CrossRef]
22. J. H. Alkhateeb, M. Alseid, "DBN – based learning for Arabic handwritten digit recognition using DCT features", In: 2014 6th International Conference on Computer Science and Information Technology (CSIT), pp. 222—226 (2014). [CrossRef]

23. M. Salameh, "Arabic digits recognition using statistical analysis for end/conjunction points and fuzzy logic for pattern recognition techniques", World Comput. Sci. Inf. Technol. J. 4(4), 50—56 (2014).
24. O. E. Melhaoui, M. E. Hitmy, F. Lekhal, "Arabic numerals recognition based on an improved version of the loci characteristic", Int. J. Comput. Appl. 24(1), 36—41 (2011). [CrossRef]
25. P. P. Selvi, T. Meyyappan, "Recognition of Arabic numerals with grouping and ungrouping using back propagation neural network", In: International Conference on Proceedings of Pattern Recognition, Informatics and Mobile Engineering (PRIME), pp. 322—327 (2013). [CrossRef]
26. S. A. Mahmoud, "Arabic (Indian) handwritten digits recognition using Gabor-based features", In: International Conference on Proceedings of Innovations in Information Technology, IIT 2008, pp. 683—687 (2008). [CrossRef]
27. A. El-Sawy, H. El-Bakery, M. Loey, "CNN for Handwritten Arabic digits recognition based on LeNet-5", Springer International Publishing AG 2017. [CrossRef]

AUTHORS PROFILE

Hamdy Amin Morsy (MSc 2002, PhD 2012) is an assistant professor at Helwan University, Faculty of Engineering, Electronics and Communications Engineering Department, Cairo, Egypt. He is also a visiting assistant 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 in 2012.