Nigerian Research Journal of Engineering and Environmental Sciences 6(1) 2021 pp. 293-298

Nigerian Research Journal of Engineering and Environmental Sciences
Journal homepage: www.rjees.com

Original Research Article

Multi-Layer Perceptron Neural Network for an Offline Signature Verification System

***[1]Nuhu, A.S., [2]Adam, N., [2]Gadam, A.M. and [1]Dajab, D.D.**

[1]Department of Electrical and Electronics Engineering, University of Jos, Plateau State, Nigeria.
[2]Department of Electrical and Electronics, Federal Polytechnic, Bauchi, Bauchi State, Nigeria.
*nuhua@unijos.edu.ng

ARTICLE INFORMATION

Article history:
Received 17 Feb, 2021
Revised 23 Apr 2021
Accepted 03 May 2021
Available online 30 Jun, 2021

Keywords:
Feature extraction
Segmentation
Image processing
Multi-perceptron
Recognition

ABSTRACT

Signature verification using neural networks is characterized by the use of pre-processing techniques such as normalization, morphological operations and median filtering. In this work, an effective method for offline signature verification system based on multi-layer perceptron (MLP) was proposed. A signature can be divided into five logically connected, basic aspects or layers which are learnt by a single set of weights. The system was built based on a four-hidden layer neural network. An accuracy of 82.5% was attained in recognizing genuine and forged signatures which outperformed the state-of-the-art techniques that incorporate feature selection and preprocessing operations.

1. INTRODUCTION

A signature is regarded as behavioural biometric of a person to show his/her ideal identity on printed documents. Areas of authorization based on signature include credit card validation, security systems, banking system, checks, contracts, etc. It is commonly used as evidence of self-identity and a granted access technique. The domain contact individual are personnel, companies or banks that required to validate their signatures. Therefore, a contact personnel from organizations such as banks may need authentication in order to have access to their database, and bank's staff also need to validate if the sample signature is genuine in order to finalize a transaction. A framework for offline signature verification system was reported in Sharif *et al.* (2018) and Chandra and Maheskar (2016). Development of offline handwritten signature authentication system using artificial neural network was reported by Gunawan *et al.* (2017) and Rahmi *et al.* (2016). Although their system reported faster convergence time but low accuracy was reported. Similarly, signature recognition and verification system using back propagation neural network was developed by Choudhary *et al.* (2013). Offline signature verification through probabilistic neural network was proposed in Yin *et al.* (2010) as well as Odeh and Khalil (2011). But the problem with these techniques is lower accuracies, trial and error method in selecting the best features, and complex pre-processing operations.

Current works in the field of signature verification using neural networks are characterized by employing the use of pre-processing techniques like normalization, morphological operations and median filtering. Above all, the use of feature extraction to derive the most relevant and best features in the data as a trick for improving accuracy at the same time, bringing down the training time. In this work, a method for offline signature verification that doesn't need the time consuming trial and error work of determining which set of features work best by using different feature selection algorithms along with pre-processing operations, is proposed. The method is based on the selection of the right neural network architecture which follows from the application of one of the fundamental concepts in neural networks; each layer learns a certain aspect in the data.

A signature can be divided into five logically connected, basic aspects or layers which can each be learnt by a single neural network layer (or in a more technical term, a single set of weights). As such, employing a four-hidden layer neural network together with minimization of each input data size to bring down training time, is all needed to achieve high enough acceptable accuracy in verifying signatures. The task of verifying signatures is a complex one, therefore the need for automated and accurate MLP artificial neural networks system is necessary. This paper proposes an approach that used end-to-end neural network architecture to learn and recognize patterns required in verifying a signature without the need for intervention of the system designer.

2. MATERIALS AND METHODS

2.1. Model Training Procedure

The implementation framework was based SigCom dataset which consists of collected images of genuine signatures and their corresponding possible forged forms Liwicki, M. *et. al.* (2012). Moreover, it is freely available unlike many others which involve long processes before one can access the dataset. Next, pre-processing of the images was performed to remove the noise in the images, then followed by feature extraction, feature representation then learning and classification of genuine or forge signature. The software used in development of the system was WingWare Python IDE and the Anaconda IDE along with the required library modules. The choice for the dataset was due to the format and annotation that completely fits the problem being addressed. Furthermore, data augmentation was performed to improve system performance. To achieve that, two additional images were generated for each signature image by rotating the original image by 10 degrees clockwise and anti-clockwise.

The following steps were used for the system implementation:

1. The weights were randomly initialized with values between 0 and 1 each time the program is run in order to lessen the chance of getting stuck in a bad or local minimum after the training session.
2. The number of hidden layers for accomplishing the task was four (4).
3. The selection of the number of nodes in each hidden layer follows from the use of the rule of thumb that the size of the hidden layer should be somewhere between the input layer size and the output layer size or the next immediate layer size Swingler, (1996).
4. The size of the first hidden layer was set to 13. The size of the second hidden layer was 7. Third hidden layer is 5 and the fourth hidden layer is 4.
5. The activation function chosen was the sigmoid function which was used due to its popularity and simplicity in use.
6. The number of output layer nodes was two (2), representing the two signature classes (genuine and forged).
7. The errors were back propagated to the hidden layers in a direct proportion to the connected weights values.
8. Gradient descent algorithm was used in the error minimization process and the subsequent weights update.

Determining the values of the learning rate and the epochs is a matter of trial and error along with observing a trend in the performance/accuracy of the system as the values are being changed.

2.2. System Framework

The selection of the right MLP architecture, follows from the application of fundamental concepts in neural networks; each layer learns certain aspect in the data. From just a quick examination and analysis, a signature can be divided into five logically connected aspects or layers which can be learnt by a single neural network layer (Figure 1). At the first hidden layer, the network learns the particular pixels that form a signature and distinguished from the other pixels that form the image background. At the second hidden layer, the network learns how the pixels were identified in the previous layer combine to form edges of a signature. At the third layer, the network learns the orientation of the edges and form generalizations about the possible different orientations for a given data category. At the fourth layer the network learns how the oriented edges learnt from the previous layer, combine to form a whole signature. And finally at the output layer, the network learns how to categorize a whole signature as either genuine or forged. The Figure 1 would seem sufficient in carrying out the desired task but in practical sense, it is known that the more the hidden layers and hence, the more the hidden nodes, the harder it is to train the network (i.e. the more the training data required) and the more would be the program run-time. The solution to this problem is reducing the resolution of the signature images to 5 pixels × 5 pixels such that the number of input nodes of the network is only 25. Therefore, it follows that subsequent layers would have less and less number of nodes since the work of each layer is indirectly nothing more than representing the features in the previous layer in a smaller form. Each node in any layer is forward connected to all nodes in the next layer. Notice from the diagram that some connections were intentionally left out just to avoid jumbling of the diagram.

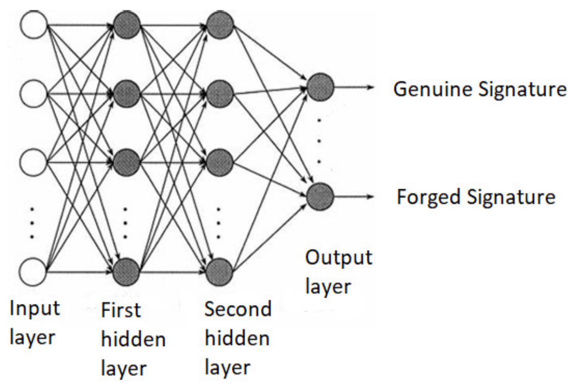

Figure 1: The two-hidden-layer MLP neural network

The first set of nodes arranged vertically in the diagram, form the input layer. These nodes are in real sense, nothing more than the pixel values that form the signature image being fed into the system. These inputs are then further fed into the first hidden layer depending on the values of the link weights connecting the input layer and the output layer. At each of the first hidden layer nodes, the signals are arithmetically summed and an activation function (the sigmoid function chosen in this work) is applied on the sum to produce outputs of this layer. The inputs continue their journey in this fashion up to the output layer where upon application of the sigmoid activation function, the output of the network is finally obtained which represents the network's answer. This phase is called the forward propagation. The next step that follows is the calculation of errors obtained by comparing the expected or target output values and what the network actually output. These errors are then back propagated into the hidden layers and finally, the link weights are updated in such a direction as to minimize the error using the gradient descent algorithm. And thus, learning is said to have taken place. The form of learning used in this neural network is called supervised learning meaning that the network needs to be trained first on examples before it can learn how to accomplish the task, as opposed to unsupervised learning which is beyond the scope of this work.

3. RESULTS AND DISCUSSION

At initial learning stage, it was observed that the system performance against the learning rate was constant at 50% as shown in Figure 2. After careful analysis, it was observed that more than 500 weights were adjusted for a given signature sample. Furthermore, higher performance or accuracy will be computed for a given high training time. Some weights were kept constant as the number of weight updates were too much each time a training data is passed thus causing the network to easily overshoot the true global minimum of the error function after each mass weights update. That was exactly done; the link weights that emerge from the second hidden layer to the third hidden layer were arbitrarily chosen to be kept constant without ever updating them after the first initialization of the entire link weights to randomly generated values. Thereafter, following testing of the network with 20 reserved genuine signatures and 20 reserved forged signatures. Figure 2 shows the results of the performance against learning rate for experiment without data augmentation.

Based on the outputs from Figure 3, at learning rate of 0.01, the performance was computed at 67.50%. Unlike the other cases where either the network believes they are all genuine or all forged with high confidence; meaning the network is not on the right track to learning anything. Therefore, it is computationally complex for the model to take all input as genuine or everything it sees as forged. Thus, the best or optimum learning rate was achieved at the learning rate of 0.01. Comparison with the state-of-the-art methods is presented in Table 1.

Table 1: Performance of different signature verification techniques and approaches

Technique/Author	Performance (%)
Proposed technique	82.50
Baltzakis and Papamarkos, (2001)	80.81
Chandra and Maheskar, (2016)	80.00
Odeh and Khalil, (2011)	78.80

The result obtained for the proposed method, 82.50%, is higher than the 80.81%, 80.00% and 78.80% recorded in the work of Baltzakis and Papamarkos, (2001), Chandra and Maheskar (2016) and Odeh and Khalil, (2011) respectively. The variation in accuracy could have resulted from the learning rate, feature selection or noise presence in the row images.

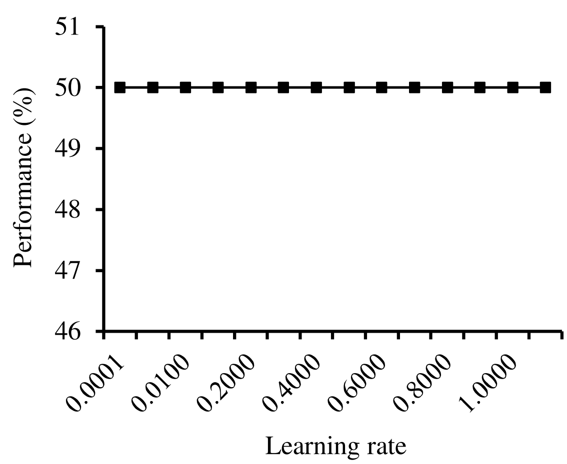

Figure 2: Performance versus learning rate without data augmentation

The system performance was obtained as epochs varied and learning rate maintained at the optimal value as presented in Figure 2 i.e. learning rate = 0.01. Furthermore, the network starts over-fitting as presented in Figure 3 when the number of epochs exceed 14,000 and therefore, the learning stopped. Performance curve presented in Figure 4 showed the accuracy obtained 82.50% at learning rate of 0.01.

A.S. Nuhu et al. / Nigerian Research Journal of Engineering and Environmental Sciences
6(1) 2021 pp. 293-298

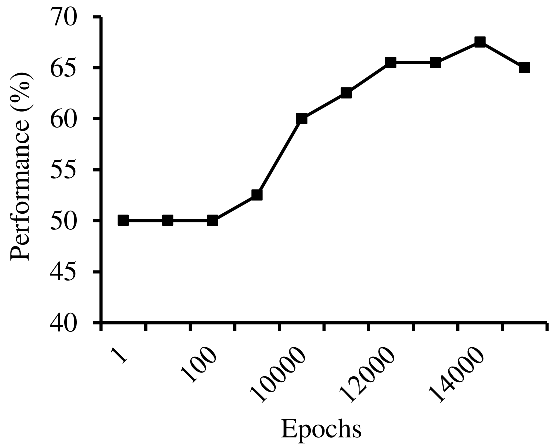

Figure 3: Performance versus epochs with data augmentation

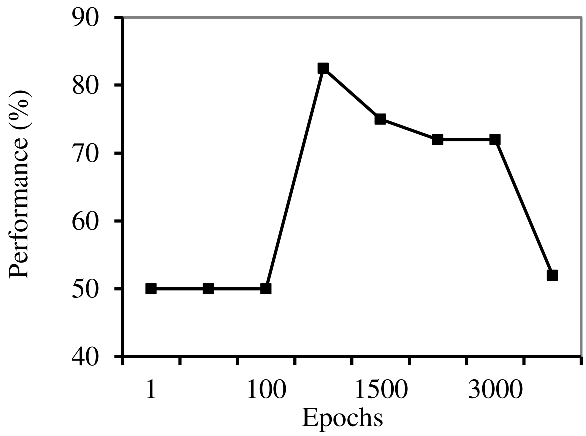

Figure 4: Performance versus epochs with data augmentation

4. CONCLUSION

In this work, performance computation of 82.5% at a learning rate of 0.01 was achieved. It was evident that signature verification with acceptable level of accuracy was obtained using multi-layer perceptron neural network. The NN architecture together with data augmentation were used in the model training. The performance of roughly 80% was acceptable for a signature verification because in banks for example, a customer writes his/her signature more than once for the purpose of validating and recognition of correct identity of the enrollees on the database.

5. ACKNOWLEDGMENT

The authors wish to acknowledge the assistance and contributions of Department of Electrical and Electronics Engineering, University of Jos and Department of Electrical and Electronics Engineering Federal Polytechnic Bauchi, toward the success of this work.

6. CONFLICT OF INTEREST

There is no conflict of interest associated with this work.

REFERENCES

Baltzakis, H. and Papamarkos, N. (2001). A new signature verification technique based on a two-stage neural network classifier. *ENGAPPAI, Engineering Application of Artificial Intelligence*, 14(1), pp. 95-103.

Chandra, S. and Maheskar, S. (2016). Offline signature verification based on geometric feature extraction using artificial neural network. *3rd International Conference on Recent Advances in Information Technology (RAIT)*, pp. 410-414.

Choudhary, N. Y., Patil, R., Bhadade, U. and Chaudhari, B.M. (2013). Signature recognition and verification system using back propagation neural network. *International Journal of IT, Engineering and Applied Sciences Research (IJIEASR)*, 2(1), pp. 1-7.

Gunawan, T. S., Mahamud, N. and Kartiwi, M. (2017). Development of offline handwritten signature authentication using artificial neural network. *International Conference on Computing, Engineering, and Design (ICCED)*, pp. 1-4.

Liwicki, M., Malik M. I., Alewijnse L., Van den Heuvel E. and Found, B. (2012). Competition on Automatic Forensic Signature Verification. *International Conference on Frontiers in Handwriting Recognition*, pp. 823-828.

Odeh, S. M. and Khalil, M. (2011). Offline signature verification and recognition: Neural network approach, *International Symposium on Innovations in Intelligent Systems and Applications*, pp. 34-38.

Rahmi, A., Wijayaningrum, V. N., Mahmudi, W. F., Yasmin, M. and Parawe, A.M.A.K. (2016). Offline signature recognition using back propagation neural network. *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS),* 4(3), pp. 678 – 683.

Sharif, M., Khan, M. A., Faisal, M., Yasmin, M. and Fernandes, S. L. (2018). A framework for offline signature verification system: Best features selection approach. *Pattern Recognition Letters*, 139, pp. 50-59.

Swingler, K. (1996). *Applying Neural Networks*: *A Practical Guide,* 1 Ed. London: Academic Press, pp. 303.

Yin, O. S., Jin, A. T. B., Yan, H. B. and Han, P. Y. (2010). Offline signature verification through probabilistic neural network. *18th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision in co-operation with EUROGRAPHICS*, pp. 31-38.