OFFLINE SIGNATURE AUTHENTICATION: A ARTIFICIAL NEURAL NETWORK APPROACH

Riya Roy Chowdhury¹, Amber Hayat²

Assistant Professor, Department of Computer Science, HIT, Kolkata

Asst. Prof. Dept. of Computer Science, University of Petroleum and Energy Studies Dehradun Uttarakhand

*Corresponding Author's E-mail: rrchk88@gmail.com

Abstract
To avoid forgery and ensure the confidentiality of information in the field of Information Technology, security has been an inseparable part of it. In order to deal with security, authentication plays an important role. To identify the authenticated person various biometric authentication techniques are used (like iris, fingerprints, palm vein, signature authentication etc.). That measures physiological or behavioral characteristics such as a signature or a voice sample. Offline signature deals with the signature image acquired by a scanner or a digital camera. In this case, the handwriting order, writing speed variation, and skillfulness are the key points. Previously offline signature was verified using Hidden Markov Model (HMM), Support Vector Machine (SVM) or using some unique features of a signature. Some unique feature like global feature like pixel density, pixel distribution and pixel axil or mask feature, grid feature. In any type of authentication technique three things are most important Accuracy, False Rejection Rate(FRR) and False Acceptance Rate(FAR). In this paper we deal with off-line signature and verification with Artificial Neural Network (ANN) using back propagation neural network and obtained satisfactory results when compared with existing approaches. Here after comparing target and predicted output, the error is calculated that is always less than 0.05.

Keywords: ANN, FRR, FAR.

1. Introduction
Humans recognize each other on basis of their various behavioral characteristics. Identity verification (authentication) in computer systems has been traditionally based on something that one
has (key, magnetic or chip card) or one knows (PIN, password). Things like keys or cards, however
tend to get stolen or lost, and even passwords are often forgotten or disclosed. To identify the
authenticated person biometric authentication techniques are widely used. These characteristics are
not being duplicable but sometime unfortunately create a copy of the true sample[1-2]. Signature
authentication technology uses the dynamic analysis of a signature to authenticate a person. Hand
written signature or offline signature is based on measuring writing speed, order, pressure and angle
used by the person when a signature is produced. No encryption or message confidentiality offered
with signature authentication technique more modern examples use one-way hash functions to encrypt the signature and data to append it into the document being signed. Artificial neural network
(ANN) is an important part for artificial Intelligence (AI) [3]. The human brain is capable to perform
computationally demanding perceptual acts (e.g. Face recognition, speech recognition etc.) and
control activities (e.g. Body functions and movements). Human brain is collection of more than
10 billion interconnected neurons. Each cell (neuron) is used to receive, process and transmit
information. There is two part of each neuron Dendron and Axon [4]. ANN is generalizations of
mathematical models of biological nervous systems. In the place of neuron use nodes and they are
connected through a path like Dendron have some value called weight. In this paper two signatures
are matched using back propagation algorithm of artificial neural network. Three most widely used
ANN learning strategies are 1) error back-propagation, 2) Kohonen, and 3) counter propagation. ANN
is effectively used in solving ‘unknown’ or ‘unsupervised’ problem. Among various learning
mechanisms two processes are mostly used supervised and unsupervised learning. In the supervised
learning process network is provided with both input and output. This output is called target output.
Then the networks process the input and produce the output. This is called predicted output. Errors
are the difference between target and predicted output. Back propagation neural network is
belonging into supervised learning category. Each and every time propagated into backward and the
networks again calculate the error until the value is minimum. In other hand unsupervised learning is
also called Adaptive learning. In this type of learning process only input is provided [5]. The system
decides what feature it will choose to group the inputs and produce an unexpected output. This type
of learning process is used where system will learn from environment, example science fiction type
of Robots. They continually learn from the environment in won and they encounters new situation.

In neural network several algorithms are used but we choose back propagation algorithm.
Because of it’s preserving efficiency and easy to implement. Back propagation algorithm is used in
our work; this algorithm requires at least two layers of neurons. The input layer and output layer.
Suppose there is ‘r’ input and ‘T’ is target output. The error back propagation method has obtained
its name due to its learning procedure .the weights are first corrected in output layer then second
hidden layer and at the end in the first hidden layer. After that the weight is obtains the signals
directly from the input.

2. Related Works

A number of biometric methods have been introduced, but few have gained wide acceptance.
Before we discuss about our approach, we have concentrated on some techniques to gather
knowledge.

2.1 Hidden Markov Model

An offline signature is verified using three features of a signature are pixel density, distribution
and axial slant feature of Hidden Markov Models. Sometime offline signature are verified using intrapersonal and interpersonal variability based on HMM. EDSON, FLAVIO & ROBERT make group of
signature for 40 sample of each 40 writer and another group contains 40 signature of each 60 writers.
First group contain 1600 genuine signature sample called Intrapersonal and second group contain
2400 signature called interpersonal. Here they work with ‘geometric’ based feature mean they focus
into static features [7-8]. That they cannot describe adequately the motion of writing. The first
database creates codebooks. The system doesn’t calculate small differences between genuine signature and a test signature [9].

2.2 Support Vector Machines

In a signature verification technique verification and recognition is done by using global direction and grid features of signatures. For this signature verification technique Support Vector Machine (SVM) was used. SVM method was introduced by V. Vapnik et al. SVM method learn linear threshold functions in nonlinear case. Support Vector Machines is suitable to work on static feature of a signature, i.e. geometrical properties. Luis E. Martinez, Carlos M. Travieso, Jesus B. Alonso. Miguel A. Ferrer[10] calculate a hyper plane, that separate the data into difference classes .They follow four parameterization techniques to classified the data, i.e. Contour Measure, Counter Following ,Region Grouping  and Direct Image. Different types of errors are calculated using this technique. Using SVM they first they searching for best technique and after that they searching for best Kernel. This technique is deals with static features of a signature but never deals with motion, speed of handwriting. In an another experiment Emre Özgündüz, Tülin Şentürk and M. Elif Karslıgil* verify signatures using SVM. They divide the total system into two major parts 1) Training signature 2) Recognition or Verification of signatures. In training section they preprocess signatures using four stapes A) Background Elimination B) Noise 1)Extract Feature  2)Mask Feature  3)Global feature and 4)Grid Feature. 1320 signatures are used of 70 persons.

2.3 Receiver Operating Characteristics

Honno and Robert[12] develop a signature verification system using Receiver Operating Characteristics (ROC). They create a data set of 765 signature(432 genuine and 333 skilled forgeries ) from 51 writers. In this technique there are four possible outcomes. If a genuine signature is classified as genuine or true it is “true positive” and if it is classified as false then outcome “true negative” and if forgery signature is classified as genuine then it is “false negative” or if forgery signature is classified as false then it is “false positive”. All four types of classifications are denoted as T+ , T –, F- and F+. True positive rate (TPR) = T+/ (T+ + F-) and false positive rate (FPR)=F+/ (T- + F+). In all the previous method they use either a big database or they cann’t deals with static property. Otherwise their success rate is less. In our method we try to overcome all this difficulties.

3. Proposed Methodology

3.1. Overview

Humans are recognized using their unique features .Biometric authentication technique is one of those techniques. To identify the authenticated person biometric authentication techniques like signature authentication, fingers print, iris etc. are mostly used. That measures physiological or behavioral characteristics such. Offline signature just deals with signature image acquired by a scanner or a digital camera. In this case, the handwriting order, writing speed variation, and skillfulness are the key points. Previously many technologies are introduced to verify signatures. Some of them like HMM, SVM, ROC are accepted for signature authentication purpose. In our experiment we use back propagation algorithm of artificial neural network.

3.2. Proposed Technology Design
3.3 Algorithm

In ANN among various learning mechanisms two processes are mostly used supervised and unsupervised learning. In supervised learning procedure network is provided with both input and output. The given output is target output and calculated output is predicted output.

Step 1: There is three layers present in the network; Input layer (I), Output layer (O) and between Input and Output Layer there is another layer called Hidden Layer (H).

Step 2: Connects the neurons of Input Layer to Hidden Layer and Hidden Layer to Output Layer with dedicated path called Weight (W).

Step 3: Initialize the path with some value. It is proved that the Neural Network works better if the initial weight is lie between -0.5 to + 0.5.
Step 4: By using Linear activation function Output of Input Layer is evaluated. This output is treated as the Input of Output Layer.

Step 5: Computes the Input to the Hidden Layer by multiplying corresponding weights of synapses. The output of the Hidden Layer is the input for Output Layer. Hidden Layer units calculate the output using Sigmoid functions.

Step 6: Calculate the output of Output Layer unit.

Step 7: Calculate the error.

Step 8: Propagate that into backward direction. Again calculate the error.

Step 9: Repeat the previous steps until the error rate is less than the target value.

BPNN—Back propagation (Rumelhart and Mc. Clelland, 1986) is used for layered feed-forward Artificial Neural Network (ANN). Because ANN is a layered architecture. The neural network send the input signal “forward”, and then the errors are propagated to “backward”. The network receives the input from input layer, the output is given in output layer and the total calculation is done into hidden layer. There maybe one or more had hidden layers. The Artificial neural network support supervised and unsupervised learning procedure. Supervised means we provide the algorithm with example of the input and output we want that the network compute the input and the error (difference between the target or actual and predicted or expected output) is calculated. The back propagation algorithm reduces error until the neural network learns the training data. The training starts with random weights (-0.5 to +0.5) and aim is adjust the weights that the error will be minimum. Normalize the input and outputs with respect to their maximum values. It is proved that the neural networks work better if input and output lie between 0-1. For each training pair, assume there are ‘l’ inputs given by ll and ‘n’ outputs on in a normalized form. If the number of neurons (let m) in the hidden layer to lie between l<m<2l. [V] represents the weights of synapses connecting input neurons and hidden neurons and [W] represents the weights of synapses connecting hidden neurons and output neurons. Initialized the weights to small random values usually from -0.5 to +0.5. For general problems, can be assumed as 1 and the threshold values can be taken as zero.

\[ [V]^0 = [\text{random weights}], [W]^0 = [\text{random weights}], [\Delta V]^0 = [\Delta W]^0 = [0]. \]

Activation function of an ANN is a weighted sum (sum of input \( x_i \) multiplied by there respective weights \( w_{ji} \)).

\[ A_j(x, \bar{w}) = \sum_{i=0}^{n} x_iw_{ji} \]  

The output function is two types linear and Sigmoid function. If the output is equal to activation function then that is called linear and the most common output function is called Sigmoid function.

\[ O_j(x, \bar{w}) = \frac{1}{1 + e^{-A_j(x, \bar{w})}} \]  

For the training data, present one set of inputs and outputs. Present the pattern to the input layer. By using linear activation function, the output of the input layer may be evaluated as

\[ \{I\}_I = \{O\}_0 \]  

computes the inputs to the hidden layer by multiplying corresponding weights of synapses as

\[ \{I\}_H = [V]^T\{O\}_I \]  

\[ m \times 1 \quad m \times 1 \quad l \times 1 \]

The hidden layer units evaluate the output using the sigmoid function as

\[ \{O\}_H = \frac{1}{1 + e^{-\{I\}_H}} \]  

computes the inputs to the output layer.

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)  
Universal Scientific Organization, www.aeuso.org  
PISSN: 2411-6173, EISSN: 2305-0543
\{I\}_O = [W]^T \{O\}_H \tag{6}
\begin{array}{ll}
n \times 1 & n \times m \quad m \times 1
\end{array}

The hidden layer units evaluate the output using the sigmoidal function
\{O\}_O = \frac{1}{1+e^{-\{I\}_O}} \tag{7}

The Sigmoid function is very close to 0 or 1 for large positive number and 0 for large negative number that creates a smooth transition between low and high outputs. Now, the aim of the training process is obtain the predicted or desire output when inputs are given. The error is the difference between the target and the predicted output.
\begin{equation}
E_p = \frac{1}{n} \sum_{i=0}^{n} (T_i - O_i)^2 \tag{8}
\end{equation}

We take the square of the difference to get a positive output. How much the difference is big the value will be greater or the difference is small the value will be smaller. Total error of the network is
\begin{equation}
E(\vec{x}, \vec{w}, \vec{d}) = \sum_{j=0}^{m} (O_j(\vec{x}, \vec{w}) - d_j)^2 \tag{9}
\end{equation}

Using gradient descendent we can adjust the weight because the error depend on the input weight and output
\begin{equation}
\Delta w_{ji} = \eta \frac{\delta E}{\delta w_{ji}} \tag{10}
\end{equation}

When we get error ‘E’ with respect to \(W_{ij}\) the goal of back propagation algorithm is propagate the error to backwards . First we derive the error E in respect to \(O_i\) . Now adjustment of the weight (\(\Delta w_{ij}\)) will be negative ,constant eta(\(\eta\)) multiplied by the dependence of previous weight . The adjustment of the weight will depend on \(\eta\). \(d\) is a constant.
\begin{equation}
[d] = (T_k - O_{ok})O_{ok}(T_k - O_{ok}) \tag{11}
\end{equation}

\begin{equation}
[Y] = \{O\}_H (d)
\begin{array}{ll}
m \times n & m \times 1 \quad 1 \times n
\end{array}
\end{equation}

Therefore the adjustment of each weight is
\begin{equation}
[\Delta W]^{t+1} = \alpha [\Delta W]^t + \eta[Y] \tag{12}
\end{equation}

Now we adjust the weights of hidden layer to Input layer we need to change was ‘V’ and calculate ‘d*’ in the place of ‘d’.
\begin{equation}
[e] = [W] \quad \{d\} \tag{13}
\end{equation}

\begin{equation}
\{d\} = e_i(O_{HI})(1 - O_{HI}) \tag{14}
\end{equation}

Find
\begin{equation}
[\Delta V]^{t+1} = \alpha [\Delta V]^t + \eta[Y]\tag{15}
\end{equation}

Find
\begin{equation}
[\Delta V]^{t+1} = [\Delta V]^t + [\Delta V]^{t+1}\tag{16}
\end{equation}

Find
\begin{equation}
[\Delta W]^{t+1} = [\Delta W]^t + [\Delta W]^{t+1}\tag{17}
\end{equation}

Find error rate as
\begin{equation}
\text{Error rate} = \frac{\Sigma E_p}{n \text{ set}} \tag{18}
\end{equation}

Algorithm : BACKPROPAGATION ALGORITHM
Here another five algorithms are discussed which are required for this experiment. I) calculation of grey value II) calculate random number III) Transpose of a matrix IV) multiplication between two matrices V) sigmoid output of a matrix.

4. Results

4.1 Signature Database
Take signatures of a person and store it in our database as original signature org_sig. Take some other signature as sample signature. In our experiment we use three sample signature called sam_sig.1, sam_sig.2 and sam_sig.3 in bmp format. Calculate grey scale value of org_sig.bmp without using Neural Network and store it. This is target value. Now take one sample signatures and Break the pixels of the image into its RGB values. Then Create a weighted Neural Network with three input nodes and one output node and Assigned value for each path of the network. The value will be -0.5 to +0.5. This is a maltly layer network with one input layer, one hidden layer and one output layer. R, G and B are the three input node for the network and gray scale value will be the output node. Train the weights with the above training set until the difference between output value of Sample Signature(sam_sig.bmp) and Original Signature( org_sig.bmp ) grayscale value is minimum. store the Weight Matrix(wt_mat1). Repeat it for another two signature (sam_sig2.bmp, sam_sig3.bmp). Similarly store Weight Matrix(wt_mat2, wt_mat3). Now calculate average weighted matrix (wt_mat1 + wt_mat2 + wt_mat3)/3. Make a new network using this average weighted value. Input the pixel values of the original signature into this network. get a output value called predicted value, if there is difference between predicted value and target values is more than 0.5 the signature is not authenticated.

Figure 4: Original Image org_sig.bmp

Figure 5: Sample Images sam_sig1.bmp, sam_sig2.bmp, sam_sig3.bmp

4.2 Evaluation
Here three tables are included for analysis purpose for three different signatures. Height 1.5” and Width 1.25”. From table [1] we get Number of iterations each pixel for training weight matrix : 7421 times Input file for target: original signature org_sig.bmp. Input file for prediction : sample signature sam_sig1.bmp, sam_sig2.bmp, sam_sig3.bmp.
After 7421 times iteration of each pixel of `sam_sig1.bmp`, I get corresponding two weight matrix, between input and hidden layer (left) and hidden and output layer (right).

\[
\begin{pmatrix}
-17.0294 & -3.7524 & 18.1117 \\
-1.8606 & -2.0283 & 6.205 \\
-26.1097 & -1.8606 & 23.8842
\end{pmatrix}
\quad
\begin{pmatrix}
-3.7849 & \\
0.1231 & \\
3.7972
\end{pmatrix}
\]

After 4462 times iteration of each pixel of `sig2.bmp`, I get corresponding two weight matrix, between input and hidden layer (left) and hidden and output layer (right).

**Figure 6**: Original Signature `org_sig.bmp` (left), Sample Signature `sam_sig1.bmp` (right)

**Figure 7**: Original Signature `org_sig.bmp` (left), Sample Signature `sam_sig2.bmp` (right)
After 3923 times iteration of each pixel of sig.1.bmp, I get corresponding two weight matrix, between input and hidden layer (left) and hidden and output layer (right).

\[
\begin{pmatrix}
-11.9081 & 2.1354 & 2.0328 \\
-0.8034 & 2.4683 & -1.2099 \\
-47.0260 & 7.3918 & 8.4300 \\
\end{pmatrix}
\begin{pmatrix}
-1.8968 \\
0.1149 \\
0.0329 \\
\end{pmatrix}
\]
4.3 Experimental Results

Now the average value of the weighted matrix is

\[
\begin{pmatrix}
-13.6152 & 0.8068 & -8.9605 \\
-0.1762 & 0.9694 & 4.1674 \\
-23.7009 & 1.4404 & 13.3814 \\
\end{pmatrix}
\begin{pmatrix}
-2.5262 \\
0.1176 \\
7.9833 \\
\end{pmatrix}
\]

Now we pass the org_sig.bmp through this weighted network

Table 1

<table>
<thead>
<tr>
<th>Pixel number</th>
<th>Target output</th>
<th>Predicted output</th>
<th>Error deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>0.2275</td>
<td>0.3639</td>
<td>0.0093</td>
</tr>
<tr>
<td>28</td>
<td>0.3294</td>
<td>0.4266</td>
<td>0.0047</td>
</tr>
<tr>
<td>6283</td>
<td>0.4784</td>
<td>0.5167</td>
<td>0.0007</td>
</tr>
<tr>
<td>6284</td>
<td>0.5490</td>
<td>0.5355</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

Now calculate

\[ e = \frac{\sum_{i=0}^{n} \text{error}_i}{n} \]  \hspace{1cm} (16)

we get \( e = 0.0130 \) as a output of table[1], where \( n = 6285 \) represents number of pixels. From table [2] we get Number of iterations each pixel for training weight matrix : 3923 times.

5. Discussion

Confidentiality is one of the main issue in the field of information technology for security to avoid forgery. In this paper we discuss about a technology of signature authentication by “Back-propagation algorithm” with application. It is a new approach in the field of signature authentication. Basic algorithm of artificial neural network through Back-propagation algorithm is used here. Here we use three (Input, output and hidden) layer, six node(three in input layer, two in hidden layer and one in output layer; because number of nodes in first hidden layer is always less than or equal to number of input layer nodes and number of nodes in last hidden layer is always greater than or equal to number of nodes in output layer). There is one original signature of the authorized person and three sample signatures. When the target and predicted output is calculated, the error is always less than 0.05. After the total calculation 0.05 is our threshold value. This system gives us 99% successful rate of signature verification.

6. Conclusion

Using other techniques in previous works maximum success rate is 96.5% but in our experiment success rate is 99%. So if the signature is authorized then maybe there is some error but that is always less than 0.05. In this paper the signature authentication problem is solved through artificial neural network in further we want to try to solve this through genetic algorithm. In this process the difficulty is large amount of variation of size. Here we recognize the signatures of a particulate size. But in future try to remove these difficulties.
7. References


