Method of Combining the Degrees of Similarity in Handwritten Signature Authentication Using Neural Networks

Ph. D. Student, Eng. Eusebiu Marcu

Abstract This paper introduces a new method of combining the degrees of similarity for a hand-written signature (also known as holographic signature or biometric signature) using neural networks. This method is used for a biometric authentication system after the degrees of similarity between a signature and it's reference template are computed. The degrees of similarity are defined using Levenstein distance of the handwritten signature's features. Using this method we achieved the following biometric performance metrics: FRR: 8.45% and FAR: 0.9%.

1 Introduction

Today's security threats come in different shapes. As technology is evolving, the security threats also are evolving. Security refers to the response of a system when different types of attacks are launched on to. The current technology helps the security providers to develop new methods that are more reliable and more accurate than classic methods. One of this is the biometric security. Biometric technology or biometrics "refers to methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioral traits. In information technology, in particular, biometrics is used as a form of identity access management and access control" [1].

We will propose a new authentication method for a biometric technology (handwritten signature) and compare our method with other similar methods.

1.1 Biometric Security

There are two main biometric categories:

Ph. D. Student, Eng. Eusebiu Marcu
University Politehnica of Bucharest e-mail: marcueusebiu@gmail.com

- physiological - refers to the shape of the body
- behavioral - related to the behavior of a person

In the first category, are included DNA, iris recognition, fingerprint, hand and palm geometry, etc. In the second one are included voice recognition, gait, signature and others. Performance of a biometric system is defined by the FAR and FRR metrics. FAR or false match rate is "the probability that the system incorrectly declares a successful match between the input pattern and a non-matching pattern in the database. It measures the percent of invalid matches". FRR or false reject rate is "the probability that the system incorrectly declares failure of match between the input pattern and the matching template in the database. It measures the percent of valid inputs being rejected" [1]. These two performance metrics are the most important.

1.2 General Biometric System

All biometric systems have some components in common. These components are shown in Fig. 1 [3]:

- data capture - the biometric data is captured by a sensor
- signal processing - the biometric data is processed by a unit of code for segmentation, feature extraction and quality control
- data storage - at enrollment phase, a number of biometric samples are provided by the user of the biometric system that are stored inside a database for later use - matching process.
- matching - an user tries to use the biometric system for authentication. Therefore he/she provides a biometric sample that is match against the stored template.
- decision - occurs after the matching process and gives an answer to user's authentication request.

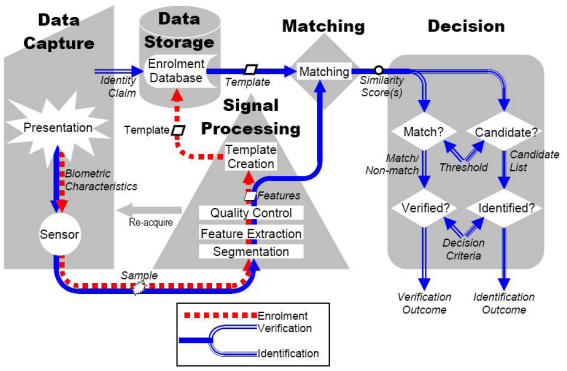

Fig. 1 Biometric system [3]

1.3 A Short Introduction to Neural Networks

Methods based on neural networks are used extensively in pattern recognition problems. One of these problems is biometric signature authentication.
The most easy way to define a neural network is that a neural network represents an attempt to simulate the brain and specific functions of living organisms. A general definition is : an artificial neural network (ANN) represents a system consisting of a large number of interconnected elementary processors (artificial neurons) that operate to solve some specific tasks. These systems have a number of advantages:

- an ANN is capable of learning;
- an ANN can operate with imprecise data;
- an ANN is capable of generalization;
- an ANN is a fault-tolerant system - if a number of neurons is lost, the network will not fail;
- an ANN can be used in modeling nonlinear systems.

The process unit inside an ANN is the neuron. This artificial neuron tries to imitate the structure and the functionality of the human neuron. An artificial neuron has a number of ***inputs***, each one having a ***synaptic weight***, an accumulator, an activation function and an output.
Therefore, we have

$$u_k = \sum w_i \cdot x_i, \tag{1}$$

The value u_k is named ***net input*** and the ***output*** of the neuron is y_k(figure 2). The function φ is called the ***activation function*** and the value θ_k is called the ***threshold*** value.

There are a number of types of learnings:

- ***supervised learning*** where the input data and the desired output is provided;
- ***unsupervised learning*** where input data and a cost function to be minimized are provided and the network provides the output;
- ***reinforcement learning***.

When training a network we must provide a learning algorithm like back propagation learning.

2 Collecting a Base of Biometric Signatures

A biometric system is measured by its performance metrics. For computing the performance metrics of a biometric system a base of biometric samples must be acquired. Our base has a number of 2932 original signatures and 2773 forgery signatures from 73 subjects. An original signature is a signature provided by a subject

and a forged signature is a signature provided by a subject A that forged other subject B having the graphical picture of subjects' B signature.

Each biometric system must have a device for capturing the biometric signals(figure 1). Our device has two 2D MEMS accelerometers [2] and an optical navigation system - ONS - that gives the relative position. Therefore, our device produces six signals (4 accelerations and 2 relative positions). These are the base signals. These signals are combined to produce new signals called ***components*** from which we extract the ***biometric features***. The biometric features are the most important part of a biometric signature. Each individual has its own biometric features for its signatures.

Based on these biometric features we compute the distance of two strings composed of biometric features using ***Levenshtein distance algorithm***. We created a number of algorithms that compares two sets of features strings that outputs the Levenshtein distance between them. Some algorithms are using only the acceleration signals, others only the ONS signals and one algorithm is using all six signals. We call these algorithms Signature Recognition Algorithms - SRA.

For a base of signatures using the SRA algorithms, we computed the distance between all signatures inside the signature base. These distances are stored in binary matrices (see table 1 - distance matrix between signature 3485 - as original - and 3480, 3481, 3482, 3483, 3484 - as samples on subject 85).

3 Types of Neural Networks

An artificial neural network contains a number of layers where each layer contains a number of artificial neurons. The most common types of neural networks are:

- feed-forward neural network
- recurrent neural network
- Kohonen self-organizing network

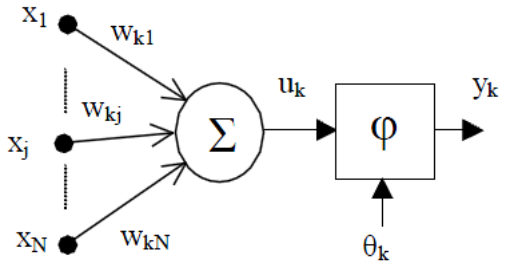

Fig. 2 An artificial neuron

Table 1 Distance matrix for signature 3485 vs. 3480, 3481, 3482, 3483, 3484

Sign id	3480	3481	3482	3483	3484
SRA1	0.667383	0.638186	0.690937	0.696175	0.715465
SRA2	0.864226	0.846354	0.862053	0.845086	0.871723
SRA3	0.331611	0.48791	0.47794	0.546078	0.64832
SRA4	0.732921	0.734493	0.795157	0.653635	0.838952
SRA5	0.350115	0.380471	0.354693	0.329385	0.47436
SRA6	0.667741	0.645182	0.685711	0.689252	0.71225
SRA7	0.605706	0.596052	0.630995	0.586095	0.607647
SRA8	0.602377	0.596205	0.617832	0.616843	0.630798

- Hopfield network
- Stochastic neural networks

3.1 Neural Network Architecture

Further we'll use a total interconnected feed-forward neural network. In our experiments, we used an artificial neural network that has an input layer, a number of hidden layers and an output layer containing only one artificial neuron (figure 3).

When training the network, the output neuron gives us the average learning error for an input data and when evaluating the network the output neuron gives us the response of the network. If the response of the network is greater than a threshold value, we will say the the authentication was successful otherwise the authentication failed. Threshold value will be determined experimentally.

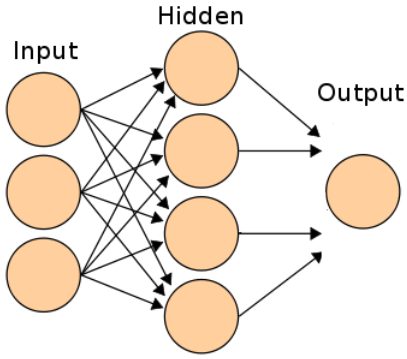

Fig. 3 An artificial neural network

4 Training and Evaluating the Neural Network

The most important thing when using an artificial network is the training data. As we said earlier (in 2), when comparing two signatures we compute the similarity of these signatures computing the Levenshtein distance.

4.1 Signature Types

As said in 2, there are two types of signatures: originals and forgeries. One subject A has a number of originals and a number of trained forgeries (see 2), forgeries by other other subjects having A's graphical signature. The original and forged signatures that are not subject's A signatures, we will consider random forgeries for subject A. We will consider five from the original signatures (picked at random) sample signatures.

Therefore, from only two types of signatures we will create four types: samples, originals, forgeries and random forgeries.

4.2 Implementing the Training and the Evaluation on a Signature Base

After collecting a base of signature, we computed a number of matrices for every SRA algorithm that contains the Levenshtein distances between all signatures.

These matrices will contain the distances between the samples and samples, samples and originals, samples and forgeries and samples and random forgeries. Using this distances, we will create the input data that trains the neural network - normalized distances. A ***normalized distance*** is given by following formula:

$$I_{M\cdot i+j} = \frac{d_{ij} - \Delta_{ij}}{\Delta_{ij}} \ , \tag{2}$$

where I is the input data array, d_{ij} is a distance in the Levenshtein matrix, Δ_{ij} is the minimum distance between samples signatures with them selfs, M is the number of sample signatures (in our tests M = 5), with $i = \overline{1,M},\ j = \overline{1,N}$ where N is the number of SRA algorithms (in our tests N = 8). The number M of signatures was determined experimentally based a classical method of authentication - the threshold method: a signature is authenticated if the average value of distances was higher than the general threshold value; if the number M was grater than 5, the result was slightly better and if it was lower then 5, the results were worst (we also noticed that subjects preferred to offer a smaller number of signatures). Therefore, the input data

is an array with $N \cdot M$ values ($N \cdot M = 40$). Each array was an input to each neuron in the first layer.

When training the network we will provide both the input data and the output data. There will be two kinds of input data and output data. The first type of input data will contain the normalized distances between the samples and originals and the output data will be an one element array contains the one (1) value. The second type of input data will contain the normalized distances between the samples and forgeries (and random forgeries) and the output data will be an one element array contains the zero (0) value. 1 stands for success (authentication succeeded) and 0 for failure (authentication failed). So, the response of the network when provided an original is the one value and when provided a forgery the response is the zero value.

In figure 4, we show the application that loads the Levenshtein distances matrices, configures the neural networks (sets the neural network parameters) and sets the signatures' exposure. The exposure is the rate of different types of signatures that will train the network. In our tests we set the 1.05 value for the originals, 2.00 value for the forgeries and 3.00 value for the random forgeries. This means that the network will learn more about forgeries than originals.

When building the input data, we pick at random the signature (in fact it's the Levenshtein distance) that will create an element inside the input data and it is pos-

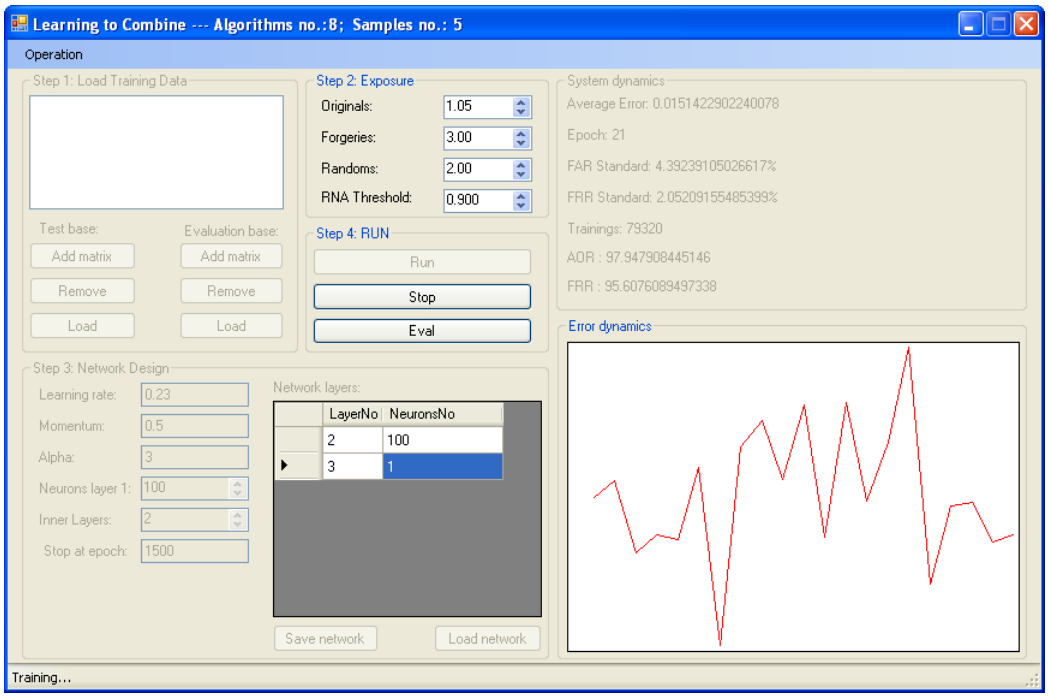

Fig. 4 Neural network application

sible to pick the same signature again. This is like when teaching a baby to learn some basic words: repeating the word over and over again.

4.3 Results Obtained

Using this procedure, we obtained the results - see table 2 and table 3. In these tables have show the best results together with the network configuration that gave these results.

Table 2 Results Obtained - architecture 1

Parameter	Value
Learn rate	0.23
Momentum	0.5
Alpha	3
Threshold	0.9
Original exposure	1.05
Forgery exposure	3
Random exposure	2
Neurons in layer 1	100
Neurons in layer 2	100
Neurons in layer 3	1
Epoch	120
FRR	8.4451460142068%
FAR	0.948692956032%

Table 3 Results Obtained - architecture 2

Parameter	Value
Learn rate	0.18
Momentum	0.023
Alpha	3
Threshold	0.9
Original exposure	1.05
Forgery exposure	3
Random exposure	2
Neurons in layer 1	5000
Neurons in layer 2	1
Epoch	700
FRR	9.5501183898974%
FAR	0.9191427802746%

4.4 Comparisons with other similar methods

Similar methods are using also a feed-forward neural network but also a RBF neural network. The table 4 contains the values of the FAR and FRR performance metrics.

Table 4 Comparison with other similar methods

Method name	FAR value[%]	FRR value [%]
Neural Network-based Handwritten Signature Verification [5]	2.0	1.3
Off-line Handwritten Signature Verification using Radial Basis Function Neural Networks [6]	4.89	6.94
Off-line Signature Verification Using Fuzzy ARTMAP Neural Network [7]	7.27	11
On-line Signature Biometric System with Employment of Single-output Multilayer Perceptron [8]	6.45	0

5 Conclusions

The paper introduces a method based on neural networks for authenticating handwritten signatures. In this paper we tried to minimize the FAR value and we considered only the values below 1%. In our tests we trained and evaluated diffrent types of NN architectures (increasing the number of hidden layers and the neurons inside them) but the best results were given by the 100-100-1 architecture.

The future works involves the creation of new algorithms SRA better defining degrees of similarity between original and forgery signatures. Obviously, this will influence neural network architecture and range of training data. Having a more precise way to separate originals from forgeries, we will train the network with fewer data and the evaluation response will be much better.

References

1. http://en.wikipedia.org/wiki/Biometrics
2. (WO/2006/085783) SYSTEM AND METHODS OF ACQUISITION, ANALYSIS AND AUTHENTICATION OF THE HANDWRITTEN SIGNATURE, http://www.wipo.int/pctdb/en/wo.jsp?IA=WO2006%2F085783
3. International Committee for Information Technology Standards, 2007, Study Report on Biometrics in E-Authentication, 30 March 2007, INCITS M1/07-0185
4. Bai-ling Zhang, Min-yue Fu, Hong Yan.: Handwritten Signature Verification based on Neural 'Gas' Based Vector Quantization. 1051-465l/98 IEEE (1998)

5. McCabe, A., Trevathan, J., Read, W.: Neural Network-based Handwritten Signature Verification, JOURNAL OF COMPUTERS, VOL. 3, NO. 8, AUGUST 2008
6. Azzopardi, G., Camilleri, K.P.,: Offine Handwritten Signature Verification using Radial Basis Function Neural Networks
7. Murshed, N.A., Bortolozzi, F., Sabourin, R.: Off-line Signature Verification Using Fuzzy ARTMAP Neural Network, 0-7803-2768-3/95 IEEE 1995
8. Szklarzewski, A., Derlatka, M.: On-line Signature Biometric System with Employment of Single-output Multilayer Perceptron, Biocybernetics and Biomedical Engineering 2006, Volume 26, Number 4, pp. 91102
9. Radhika,K.R., Venkatesha, M.K., Sekhar, G.N.: Pattern Recognition Techniques in Off-line hand written signature verification - A Survey, PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY VOLUME 36 DECEMBER 2008 ISSN 2070-3740
10. Katona, E., Kalman, P., Toth, N.: Signature Verification Using Neural Nets