Algorithms selection and adaptation in accord with architecture for RBF neural network based face authentication SoC

Lionel Pierrefeu, Jacques Jay

Algorithms selection and adaptation in accord with architecture for RBF neural network based face authentication SoC.

Lionel Pierrefeu
Laboratory Hubert Curien - IMAGe
University Jean Monet
18, rue B. Lauras
FRANCE
lionel.pierrefeu@univ-st-etienne.fr

Jacques Jay
Laboratory Hubert Curien - IMAGe
University Jean Monet
18, rue B. Lauras
FRANCE
jay@univ-st-etienne.fr

***Abstract**: This paper describes the algorithms applied to a Radial Basis Function (RBF) neural network. This neural network is used as a classifier to design a human face authentication system. The aim of this project is to obtain a low cost system on chip (SoC) to replace password identification for one person on mobile devices. Several parts of the neural network need to be modified to obtain good performances for this application with cost limited hardware resources. For the system design, the algorithms are selected and adapted in accord with architecture implantation on hardware platform (methodology AAA). Also we present different adaptations made for a full integration of RBF neural network (learning and recognition steps).*

I. INTRODUCTION:

Authentication and recognition systems based on biometric measurements are in full development because of their potential applications in many research areas, such as surveillance or security for access control or personal identification. Some systems rely on stable characteristics of the body such as fingerprints, hands or iris geometry, other exploit dynamic features such as voice. Human face contains a lot of details which can be used to recognize a person, and the acquisition is very easy with a simple image sensor.

Our research focuses on the development of an industrial system on chip (SoC) for face based authentication. It would replace password identification in electronic consumer's products such as cellular phones or computers. This is a challenging problem since the system must work under uncontrolled conditions (light, pose) [15]. Moreover, it has to fulfill both practical and industrial requirements. Indeed, from the user point of view, the system must be easy to use and to configure. From the industrial point of view, it has to be a real time and a low cost mono chip system.

As a result, the design conception must follow the AAA methodology (Adequation Algorithm Architecture). The chosen algorithm for face authentication is a radial basis function (RBF) neural network. This classifier has been selected for its good ratio between performances and complexity [1][2]. This neural network offers good performances and presents a very suitable architecture for process parallelization [3].

The RBF neural network configuration depends on several parameters which must be adjusted to obtain satisfying classification performances. Some of them are set during the system design. The others are determined during the learning phase and are user-dependant. So we present some solutions to perform a good parameter configuration for a limited cost. The first configuration step is partially performed by the user. It consists of acquiring several images of the user as reference images. Some tests are proposed to verify the quality of image set. The second step is the kernel configuration. We proposed an automatic solution to compute it and, at the same time, save hardware resources. Then we present simplification made for the neural network weighting configuration.

The article is organized as follows: first a general description of the RBF neural network is proposed. Secondly neural network parameters are introduced with their respective calculation solutions. Finally we present results (times, surfaces) obtained with proposed algorithms.

II. AUTHENTICATION ALGORITHM:

A. RBF neural network presentation:

In this project, the aim of an authenticator like a RBF neural network is to perform a classification between different classes exploiting their universal approximation properties. Indeed, RBF algorithm is a kernel-based network composed of several radial activation functions for interpolating [5][6][7][8] in high dimension. It allows to compute the complex decision regions by overlapping the radial functions. A training phase with a reference data set (images face of the person to identify) allows to determine the parameters of the functions. The parameters are defined with a goal of convergence towards the clustering function that leads to optimal classification results. After a short overview of the neural network principle, the RBF parameters and training method will be detailed.

B. RBF presentation:

The RBF structure [2][5][9][10] is composed of three layers (Fig. 1). Like multi-layer perceptron, each one is fully connected to the following one. The first layer of the network corresponds to the characteristic input vector (p_n). In our case, data of characteristic vectors are obtained from pixels intensity of the normalized face image. Then the input layer dimension is equal to the input data size. The second layer consists of the hearth of the network intelligence. It is used to cluster the input data. Each hidden unit performs the kernel function on the input vector. The output layer performs a weighted sum (6) of all hidden nodes.

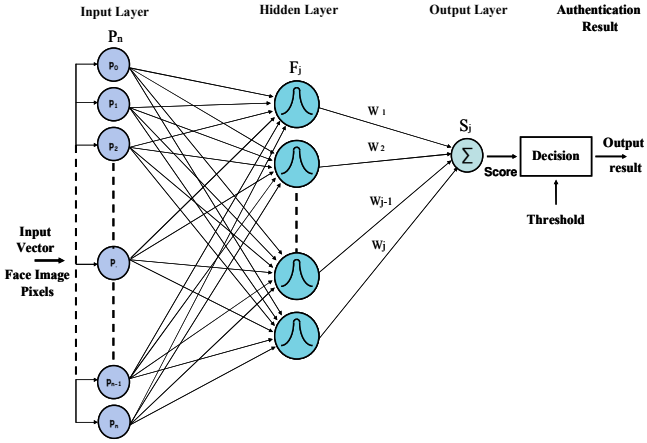

Fig. 1. Radial basis function neural network

Each hidden neuron is connected to all inputs, and compute distances (1) between references vectors and the input vector. The Euclidian distance is the most used [3][4].

$$d(c,p) = \sum_{i=0}^{n-1} \left\| c_i - p_i \right\| \qquad (1)$$

c : *reference vector*

p : *input vector*

n : *input vector size*

The kernel is an activation function filtering data (2). The closer to a reference image an input face is, the higher the result will be. This function can be interpreted as a weighted distance. For this system, we have selected the most common kernel function which is a Gaussian [3][4][9][10]. A Gaussian function is defined by two parameters: the center of the radial function which corresponds to the reference image vector and the width which corresponds to the kernel influence. The definition of RBF parameters is ensured during the training.

$$f_j(c,p) = e^{-\frac{d(c,p)}{\sigma}} \qquad (2)$$

σ : *kernel width*

As we see, each Gaussian function permits the classification cluster interpolation. The interpolation function is adjusted with learning steps. It composes of different algorithms which determine and fix each kernel parameters (Gaussian width, centroïd). So we will present in following parts each steps for the learning procedure and present all respective algorithms developed to automate this treatment.

III. ALGORITHMS OPTIMISATION:

To obtain an implantable version of the neural network, some modifications are needed. At the same time, algorithms need to be added for the performances improvement. In the following parts, we will present algorithms developed for parameters configuration.

A. Reference vectors determination:

The references vectors of kernels function are proposed by the user. It's correspond to the center of each radial function and it is used to compute the distance to the input vector. This is a crucial step which determines the performances of the system. Even with a very good classifier a good set of reference images is needed. This is the base to obtain good performances. So we propose an automatic method to help user for the reference images acquisition. Before that, we need to determine what a good set of reference images is.

Image reference set need to be acquired in condition close to usual condition of system utilization. On top of that, images need to present enough details of the object (human face). For the reference images acquisition procedure, it asks to the user to present his face in different orientations to the image sensor (see example figure 2). To obtain a good system configuration we need a lot of details of user face, it leads to have images with enough difference from one to another. It also has an importance for the definition of following parameters. So to sum up: images need to be representative and enough different to present different details.

Fig. 2. *Example of a set of face images.*

To help the user in this procedure, we have developed an algorithm which determines the quality of images set, and rejects images too similar or too far from the others images. This algorithm is simple and

does not need a lot of hardware resources and offers good results. The treatment is not based on the images contents whereas it only evaluates the distances between all images. Accepted images must respond to the test, and must have mean distance contained between two thresholds.

C. Method description:

The first step is to compute a distance between all centroids pairs (*DBCP*) to form the inter-distance matrix (3).

$$M = \begin{bmatrix} 1 & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{12} & 1 & & & \\ x_{13} & & \ddots & & \vdots \\ \vdots & & & & \\ x_{1n} & & \cdots & & 1 \end{bmatrix} \quad \text{Interdistance matrix} \tag{3}$$

For each image we cumulate all distances from others images. The result obtained is compared with two thresholds in order to determine whether the images are too close or too far from the others images. The first threshold is use to reject images too similar to others. The second one rejects images too different to others, it may detect images acquired in bad conditions. If an image is rejected, the user needs to acquire a new image.

B. Gaussian width:

As we see, the radial function selected is the Gaussian function. This function is defined with two parameters, the center of the Gaussian describes in the previous section (reference vector) and the width which corresponds to the kernel influence. The role of the kernel is to filter the distance calculated between reference vector and input vector. The Gaussian width determines the selectivity of the kernel.

For the sake of clarity, using a two dimensional representation (Fig. 3) allows to easily understand explanation of the width importance. When the Gaussian width changes it results in a variation of selectivity for the neural network. If the width is too small (Fig. 3.1), the system influence area falls and the no recognition cases increase. In inverse proportion if it is too important (Fig. 3.3), false detections appear. The selectivity of system depends on the position of each centroïd, if the distance between two reference vectors is too important, the width need to be more important. Then distances between references vectors (*DBCP*) could change the selectivity of system from employment to other and the quality replying of the authentication could be affected. So a good width calculation needs to depend on centroids position. The more a centroïd is similar to others, the more the width of this kernel needs to be influent

Generally for RBF neural network, classical clustering algorithms are employed for the calculation, like K-mean [9], other error minimization [10] and heuristic methods [11][12]. Actually all these algorithms are based on error minimization between clusters. Our neural network is a one class classifier; it has to recognize only one person face and to be able to reject each other face, so only one cluster is available. Then a solution is needed to adapt the width to an optimal value.

A solution could be the creation of a second cluster [14] with unknown person's images. It would permit to apply traditional algorithms, to calculate error minimization between clusters. From a hardware point of view, this solution is not optimal, because of important memory consumption and higher design complexity. Another solution is to keep only one cluster, and with a statistical study to develop a heuristic method based on the distance between the centroids (*DBCP*).

The Gaussian calculation is an exponential based function. It costs a lot of resources and it is a real problem to compute it in real time function. The solution is using look up table (LUT) which contains a precalculated (learning step) Gaussian function. At the same time, we decide to limit hardware cost (number of random access memory blocs) by using only one Gaussian function for all neural network kernels. So we need to find a method to calculate a unique Gaussian width which maximise performances for the entire network.

C. Proposed method:

The aim of this study is to determine an automatic method to define an optimal Gaussian width which maximizes the difference between true and false detections. With classical methods, the classification is done by interpolating a clustering function which

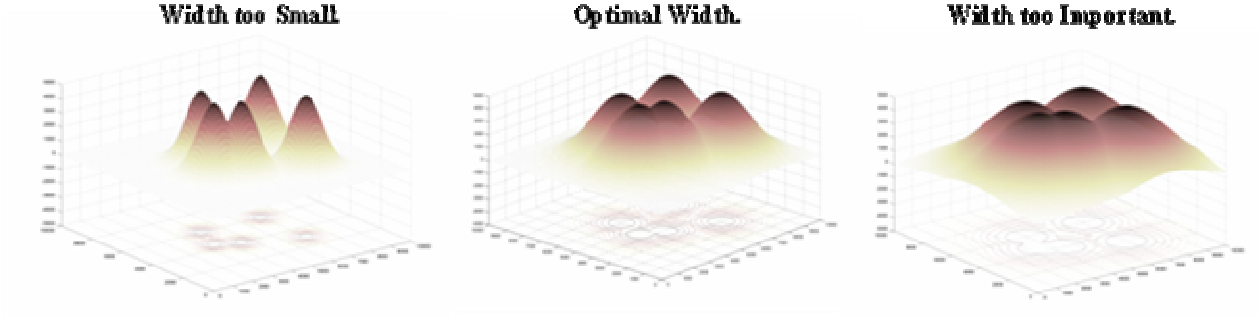

Fig.3. Neural network selectivity variation in function of selected Width

separates different classes. The optimization of the clustering function is generally performed by an error minimization between classes. As it is previously explained, in our case only one class exists. The optimal width search is performed by replacing this lack of information by using a database of face images. Testing a set of stranger faces with the neural network allows evaluating the system response and situates the false detection rate. So it replaces the error minimization function which separates the reference cluster to the stranger one, by maximizing the difference between good and wrong face authentications. Testing a lot of stranger faces permits the system to automatically converge to the optimal width.

D. *Method description:*

Referring to the relation established between the DBCP and the system selectivity, the adaptation of the width value must be based on the DBCP. If the centroids are too distant from one to another, the width value needs to be increased and reciprocally. Some tests show that referencing the width calculation with the Mean Distance Between Centroids (MDBC) gives some good results and seems to be well adapted. In fact mean distance is not the optimal value for the width. The best result is given with approximately 20% less than the mean value (4). The method proposed is simple:

- First, calculate the MDBC (4).
- Secondly, calculate the width (5).

$$MDBC = \left[\frac{2}{Nb(Nb-1)} \text{ x } \sum_{1 \le i \ne j \le Nb} \sum_{k=1}^{n} \left\| c_{i_k} - c_{j_k} \right\| \right] \quad (4)$$

Nb : *number of ref.vectors*

c : *reference vectors*

n : *input data size*

$$Width = \frac{MDBC}{Cst\_Div} \quad (5)$$

Method evaluation is done with the Olivetti Research Laboratory database (ORL) [13]. This database is composed of 10 different images of each of 40 distinct subjects. The choice of this database of face images is representative of target application. Indeed, for some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All images were taken against a dark homogeneous background with the subjects in frontal position (with tolerance for some side movement).

For the test, one series (face images of one subject) is chosen and randomly 5 images of this person are selected as reference images. These 5 references images are going to be used for the system learning and the 5 other images of this person (different from training images) are available for the test plus the 390 faces of unknown persons. This test series allows calculating detection rates (good and false). Fives tests images for the good visage are sufficient because of the low dispersion of the result and for wrong visages 395 images are enough to converge to a good result. For different values of the Cst_Div, the calculation is performed with values taken from 3.6 to 0.05 with a decrement of 0.05. The test is done with all series of the ORL database (each 40 subject becomes the reference person). Figure 4 presents statistical results obtained after calculation: the authentication results given at the output layers of the neural network. It permits to have a better visualization of the system performance. To obtain the true and false detection rates, it only requires to compared the score to a fixed threshold.

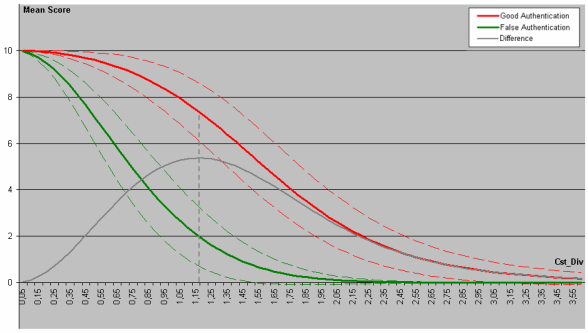

Fig. 4. Evolution of correct and false authentication with Cst_Div.

The graphic presented in the figure 4 shows the results. There are two results presented as function of Cst_Div variations: mean results for good faces images (first curve) and mean results for wrong faces images (second curve). The third curves represents the difference between the two results curves. It shows that an optimal value for Cst_Div maximizes the difference between good and false detections. The value which gives the best result is 1.2. Automatically computing Gaussian width value by referencing the width with the mean distance between centroids gives good results and optimizes the system performances. The Gaussian width calculation is automatic and adapted to images references variation.

E. *Synaptic weights calculate simplification:*

The output of the RBF neural network is computing with linear combination. The neural output consists of a weighted sum (6) as following:

$$S(f,w) = \sum_{j=1}^{m} f_j . w_j \quad (6)$$

f : *hidden layer results*

w : *synaptic weights*

Synaptic weights w_j are the weighting connections between hidden and output layers. It can be computed by searching for the linear combination which normalizes the output. The synaptic weight configurations are performed by testing an input vector and adapt weights to obtain the desired output result. The method used to achieve is a mean square minimization. To perform this operation, we need to resolve the following equation (7).

$$W = \left[\left(Y^{\mathrm{T}} Y \right)^{-1} Y^{\mathrm{T}} T \right]^{\mathrm{T}} \quad (7)$$

W : *Output weight matrix.*

Y : *Interdistance matrix M filter by kernels.*

T : *Desired output matrix.*

If the inter distance matrix is not square, the algorithm normally used for this inversion is based on Eigen vector calculation: the pseudo matrix inversion. But, the decision witch use only one cluster for the face classification allows to obtain a square matrix for inter distance matrix. So it is possible to use traditional inversion algorithm (pivot of Gauss Jordan method).

F. *System output and recognition decision:*

Classical RBF network bases their classification results on a threshold. The configuration of the threshold is very simple (binary result), but it presents a lake of flexibility. Regarding to the proposed method, varying the selectivity of the system is very easy (little variation on the width…). It could offer interesting possibility (table 1 & 2) to the system performance control (flexible to strict).

	Threshold		
Cst_Div	5	5.5	6
1.15	97.5 %	90 %	85 %
1.2	95 %	87.5 %	77.5 %
1.25	90 %	82.5 %	75 %

Fig. 5. Table of good authentication rate

	Threshold		
Cst_Div	5	5.5	6
1.15	5.0 %	5.0 %	0.0 %
1.2	5.0 %	0.0 %	0.0 %
1.25	0.0 %	0.0 %	0.0 %

Fig. 6. Table of false authentication rate

These results are obtained by the database ORL. Note that there are rapid variations of performance for a small variation of the width.

IV. RESULTS:

A. *Authentication step:*

This optimized version of RBF neural network was implemented on FPGA development board (Altera StratixII – EP2S60). The design development has been done in the respect of the AAA methodology, exploiting all parallel treatments possibilities.

Input vector is reading from an internal RAM block of FPGA. The different coefficients, reference vectors, Gaussian LUT, synaptic weights, are stack in static memory, they are preloaded from an EEPROM (SD-CARD). At the output, the input vector score is directly sent to the human machine interface (HMI). The following figure present the different parts of the network implantation.

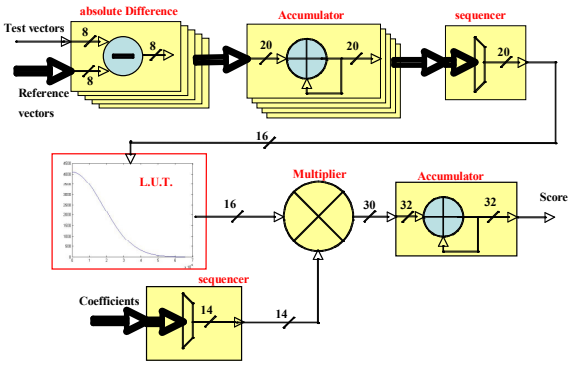

Fig. 7. Diagram of neural network

The neural network implantation (without the learning parts) uses approximately:

- 700 LUTs
- 500 registers
- 400 000 bits of memory
- one test duration time: 90µs

To resume, the implementation is very cost limited, it only use 5% of CLB and 30% internal memory resources.

B. *Learning step:*

As explained in this paper, learning step consists of compute all neural network parameters. Algorithms are writing in ANSI-C language for a soft processing on a processor IP. It does not increase the system complexity: reuse of the micro processor normally uses the communication between the system and the user. Even if it is possible to integrate it directly in VHDL IP because of their simplicity; it's a good compromise: limited in development time.

All this algorithms are simple and don't cost much time of process because of presented optimizations. Algorithms have been tested on RISC 32 bits processor Axis ETRAX 100LX (100 Mips). Treatments on this platform take less than 100ms.

With a 40 MHz processor (like Nios or Microblaze) we could estimate a time processing less than 1 second which is compatible with a standard utilization.

V. CONCLUSION:

All optimization and algorithms adaptation permits a good comprise between authentication performances, hardware resources and processing times. The proposed architecture with a soft processor allows a quick implementation of learning algorithms.

Now we still need to test the totality of this authentication system with a standard user, to verify system stability and autonomy.

VI. ACKNOWLEDGMENTS:

This work was supported by a region Rhône-Alpes research grants.

VIII. REFERENCES:

[1] N. Malasné & Al., "Localisation et vérification de visages en temps réel avec un réseau de neurones RBF", Traitement du signal, mai 2002.

[2] F. Yang and M. Paindavoine, "Implementation of an RBF neural network on embedded systems: Real-time face tracking and identity verification." IEEE transaction on neural network, Sept. 2003, vol. 14, n°5.

[3] N. Malasné, "Localisation et reconnaissance de visages en temps réel: algorithme et architecture", thesis, University of Bourgogne - 2002

[4] H. Abdi, "Les réseaux neurones", Edition PUG (Presse universitaire de Grenoble).

[5] I.Park & I.Sandberg, "Universal approximation using radial basis function networks." Neural Computations, vol.3, pp.246-257, 1991.

[6] J.Park & I.W.Sandberg, "Approximation and radial basis function networks." Neural Computation, 1993, vol.5, pp.305-316.

[7] M.Powell. "Radial basis functions for multivariable interpolation : A review." Algorithms for approximation, 1987, pp.143-167.

[8] J.Park and I.W.Sandberg. "Approximation and radial basis function networks." Neural Computation, 1993, vol.5, pp.305-316.

[9] M.T.Musawi, W.Ahmed, K.H.Chan, K.B.Faris, and D.M.Hummels. "On the training of radial basis function classifiers." Neural Networks, 1992, vol.5, pp.595-603.

[10] N. Benoudjit, C. Archambeau an al., "Width optimization of the Gaussian kernels in RBF networks." ESANN 2002 proceedings, Apr. 2002, pp. 425-432.

[11] S. C. Ahalt and J. E. Fowler, "Vector Quantization using Artificial Neural Networks Models", Proceedings of the International Workshop on Adaptive Methods and Emergent Techniques for Signal Processing and Communications, June 1993, pp. 42-61.

[12] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units", Neural Computation 1, 1989, pp. 281-294.

[13] FACE RECOGNITION HOMEPAGE: http://www.face-rec.org/

[14] R. McReady, "Real-Time Face Detection on a Configurable Hardware Platform", Université de Toronto - 2000

[15] L. Pierrefeu, A. Aubert, J. Jay et M. Courbon, "Normalisation de la luminosité d'image appliquée à un système d'authentification de visage", Read'05 INT Evry, Juin 2005