Sign Language Recognition System using Neural Network for Digital Hardware Implementation

To cite this article: Lorena P Vargas et al 2011 J. Phys.: Conf. Ser. 274 012051

View the article online for updates and enhancements.

Related content
- Digital Correlation based on Wavelet Transform for Image Detection
  L Barba, L Vargas, C Torres et al.
- Implementation of the 2-D Wavelet Transform into FPGA for Image
  M León, L Barba, L Vargas et al.
- Design and implementation of an algorithm for creating templates for the purpose of iris biometric authentication through the analysis of textures implemented on a FPGA
  F J Giacometto, J M Vilardy, C O Torres et al.
Sign Language Recognition System using Neural Network for Digital Hardware Implementation

Lorena P. Vargas¹, Leiner Barba, C O Torres and L Mattos
Optic and Computer Science Group – Popular of Cesar University,
Km 12, Valledupar- Colombia

E-mail: vargas.lorena@yahoo.com

Abstract. This work presents an image pattern recognition system using neural network for the identification of sign language to deaf people. The system has several stored image that show the specific symbol in this kind of language, which is employed to teach a multilayer neural network using a back propagation algorithm. Initially, the images are processed to adapt them and to improve the performance of discriminating of the network, including in this process of filtering, reduction and elimination noise algorithms as well as edge detection. The system is evaluated using the signs without including movement in their representation.

1. Introduction
The digital image processing is a complex task due to an image can contain large amount of information. Currently, there are several algorithms that allow performing these processes, but many of them are distinguished by the efficiency, feasibility, performance and trouble when they are implemented. Artificial neural networks have successful applications for gesture recognition and classification.

Hidden Markov models, dynamic programming and neural networks have been investigated for gesture recognition [1] with hidden Markov models being nowadays one of the predominant approaches to classify sporadic gestures (e.g. classification of intentional gestures [2]).

Fuzzy expert systems have also been investigated for gesture recognition [3] based on analyzing complex features of the sign like the doppler spectrum. The disadvantage of these methods is that the classification is based on the separability of the features; therefore two different gestures with similar values for these features may be difficult to classify.

Neural network algorithms are an option with multiple advantages, and supplementing these with hardware design tools such as FPGAs, which can reduce development time significantly. This feature allows these devices to be very useful for implementing recognition systems, in particular gesture Language.

In recent years, FPGA-based hardware systems have been used extensively for developing coprocessors, custom computing machines, and fast prototyping platforms. FPGAs are suitable for accelerating tasks that require processing of data with non-standard formats and repetitive execution of fine grain operations. A system with reconfigurable FPGA hardware has several advantages. Hardware based implementations are orders of magnitude faster than equivalent software systems that perform

¹ Lorena Vargas Quintero, Optic and Computer Science Group - Universidad Popular del Cesar.
the same task for some applications. Due to their reconfigurable nature, FPGAs can implement many different functions at different times, thus reducing the total number of components needed in a given hardware platform. New versions of design are implemented by simply downloading configuration bit streams. New functions can be added and maintenance can be performed as required. Likewise, systems can be made scalable [4].

The advances in FPGA technology have extended the capability of programmable logic to the realm of programmable system [5].

Hardware realization of NNs is an interesting issue [6], [7]. There are many approaches to implement NNs [8], [9]. The FPGA is a very useful device for realizing a specific digital electronic circuit in diverse industrial fields [10]. For example, Hikawa realizes an NN with on-chip BP learning using a field-programmable gate array (FPGA) [11], [12]. Some hardware implementations for neural network used in different applications are reporter [13]-[16].

The propose of this work is to make a hardware implementation of an neural network using Field Programmable Gate Arrays (FPGA), which is applied in gesture language pattern recognizing.

2. Characteristics of the gesture language
The sign language, or gesture, is a nature language of gesture space expression, configuration and visual perception, by means of which deaf people can establish a communication channel with their social environment, integrated for other deaf persons or anybody who knows the employed sign language. While in the oral language the communication is done in an auditive-vocal channel, the sign language has a visual gesture space.

The symbol group includes static and dynamic gestures, like gesture for the alphabet. This work employees images that represent the sign alphabet, especially the signs that do not have movement for their representation, as a first stage of the project.

- Figure 1 shows the used images for the learning of the network (47 images).

![Figure 1. Defined Gestures](image1)

3. Neural Network Design
A neural network is basically modelled as the structure shown in figure 2, in which can be observed a group of elements that interact to generate an output vector from an input vector described by the variable x. The training information is stored in the set of synaptic weight values of the neural network, and the output neuron is limited to a specific range of values of the activation function.
Output neurons can be described mathematically by

\[ y_k = \phi \left( \sum_{i=1}^{p} w_{ki} x_i + w_0 \right), \quad (1) \]

or,

\[ y_k = \phi(v_k), \quad (2) \]

where the subscript \( i \) indexes units in the input layer, \( k \) in the hidden; \( w_0 \) denotes the input to hidden layer weights at the hidden unit \( k \). An adder \( \sum \) which produces the weighted sum of inputs according to the respective weights of the connections. A activation function defines the output amplitude of that node given an input or set of inputs \( \phi(v_k) \), and \( w_0 \) is a threshold value.

A multilayer neural network was used in the design with a backpropagation algorithm. The structure of the network is formed by three layers, called the input layer, hidden layer and output layer; the basic components can be seen in figure 3 in which was used a simplified graphic notation.

For the input and hidden layer neurons were employed a hyperbolic tangent activation function with 5 neurons each one and one neurons at the output layer with a linear activation function.

3.1. Training process
During the training process of the first stage, it is used a backpropagation algorithm. The supervised backpropagation learning scheme modifies the weight in the opposite direction of the gradient of the error function to minimize a mean squared error of whole patterns, which are used to train the neural network. These algorithms build models that predict the desired values.

A gradient based algorithm, starting with an initial weight vector, estimates the error function and its gradient for training, and it is obtained a new modified weight vector. This is repeated till the error finds the set limit [17]. Therefore, by definition, the weights are updated through the expression:

\[ w^{n+1} = w^n + \alpha (-\nabla^n), \quad (3) \]

where is \( \alpha \) the learning rate of the network, and \( \nabla \) gradient of error function about to \( w^n \).
In the backpropagation algorithm is used the mean squared error that is calculated from a desired output \( d^m \) as:

\[
(e^m)^2 = (d^m - w^m \cdot x^m)^2
\]

therefore, the gradient is obtained from the error

\[
\nabla^m = -2 \cdot e^m \phi'(v^m) \cdot x^m
\]

Replacing in (3), it is obtained the following expression:

\[
w^{m+1} = w^m + 2 \alpha \phi'(v^m) \cdot x^m
\]

This process is made for all the neurons of each layer in the network.

4. Results

When the neural network is learned, the system is ready to identify and to recognize the associations stored in the correlation matrix.

To evaluate the performance of the implemented algorithm is developed a user interface in Matlab®, which permit to load the digital images to be analyzed, sent serially to FPGA, and receives the results of the analysis made by the algorithm implemented in the device.

It is used fixed images 120 x 150 pixels on gray scale with each pixel encoded between 0 and 255. As mentioned before the weights are previously loaded and stored in RAM memories.

The first process, when the input image has been stored in memory, is the binarization stage and edge detection. Figure 4 shows the result after applying these algorithms.

For the edge detection was used an algorithm of second derivative, through the Laplacian operator [18].

Due to the fact that the input of the neural network must be a vector, each test image is transformed for its subsequent analysis, this is done taking each row of the image and ordering to form the test input vector of the network.

Figure 4. a) Ideal Image, b) gray scale Ideal Image, c) Edge of the Ideal Image, d) low contrast image and poor light, e) gray scale low contrast image, f) Edge of the low contrast Image
Because of the input image has a size of 120 x 150 pixels, input vectors of 18000 elements are obtained, so each neuron of the input layer must have 18,000 weights and one threshold.

The results are discussed with reference to several configurations of the neural network, the neuron number of each layer and the internal layer numbers are modified. Eventually, the learning network is analyzed with different quantity of learning patterns.

Figure 5 illustrates the result using the network with the configuration shown in Figure 2. The neural network is learned with the images of the alphabet illustrated in figure 1, which were 47 images.

In the learning is assigned desired values for the input images, with separation among them of 0.2.

It can be seen from figure 5 that there is a closed relation between the reference sign and the obtained results. This reference line presents the desired values with which the network is learned to input image.

The network could identify 44 of the 47 learned patterns, with average performance of 94% and can recover a pattern in 60 ms. Every times are calculated with 50 MHz clock frequency. The graph indicates that there was a mistake in recovering the symbol S and T.

Similarly a Joint Transform Digital Correlator is used and the average performance achieved in the identification of the second set of images was very low, around 20%. This is mainly due to the difficulty of the correlator to discriminate patterns that have some degree of rotation and translation with respect to the original position. The type of correlator used is described in [18]. It is considered that a good average performance must be over 90% in recognizing unknown patterns, it means, that at least 25 from 27 images must be recognized. In this work, we consider that a pattern is recognized, when using a JTC, whether the correlation peak exceeds 0.8 for a normalized system.

Figure 6. Correlation between the image of the set of symbol of the figure 1a) y and the image “A” of the figure 1b)
Figure 6 shows the result of applying the correlation to compare the image of the symbol "A" in Fig. 1a (ideal set of training patterns) and the image that represents the same symbol in the second set of test patterns depicted in Figure 1b.

5. Conclusions and Future Works
The neural networks are one of the more powerful tools in the identification system and pattern recognition. The system presents a performance pretty good to identify the static images of the sign alphabetic language.

The system shows that the first stage can be useful for deaf persons or with speech disability for communicating with the rest of the people who do not know the language.

In this work, the developed hardware architecture is used as image recognizing system but it is not only limited to this applications, it mean, the design can be employed to process other type of signs. As future work, it is planned to add to the system a learning process for dynamic signs, as well as to prove the existing system with images taken in different position. Several applications can be mention for this method: finding and extracting information about human hands, which can be apply in sign language recognition that it is transcribed to speech or text, robotics, game technology, virtual controllers and remote control in the industry and others.

6. References
[1] Cracknell J, Cairns A, Ramsay C, and Ricketts 1994 Gesture recognition: an assessment of the performance of recurrent neural networks versus competing techniques In IEE Colloquium Applications of neural network to signal processing p 8/1-8/3
[2] Chambers G, Venkatesh S, West G, and Bui H 2002 Hierarchical recognition of intentional human gestures for sports video annotation. In Proc. 16th IEEE Conf. on Pattern Recognition p 1082-1085
[3] Frantti T and Kallio S 2004 Expert system for gesture recognition in terminal's user interface Expet Syst Appl 26 2 189-205
[4] Wall G, Iqbal F, Isaacs J, Xiwen L, Foo 2004 S Real time texture classification using field programmable gate arrays 33rd Applied Imagery Pattern Recognition Workshop p.131 - 135
[5] Chakravarthy N, Jzhong X 2006 FPGA-based Control System for Miniature Robots Intelligent Robots and Systems Conf. Intelligent Robots and Systems p 3399 – 3404
[6] Sheu B J and Choi J 1995 Neural Information Processing and VLSI. Norwell,MA: Kluwer
[7] Fakhraie S M Scalable closed-boundary analog neural networks IEEE Trans. Neural Network 15 492–504
[8] Reyneri L M, Implementation issues of neuro-fuzzy hardware: going toward HW/SW codesign IEEE Trans. Neural Network 14 176–194
[9] Genov R and Cauwenberghs G Kerneltron: support vector “machine” in silicon IEEE Trans. Neural Network 14 1426–1434
[10] Hauck S 1998 The role of FPGA’s in reprogrammable systems Proc. IEEE 86 4 615–638
[11] Hikawa H 1999 Frequency based multilayer neural networks with on-chip learning and enhanced neuron characteristics IEEE Trans. Neural Network 10 545–553
[12] ____ 2003 A new digital pulse-mode neuron with adjustable activation function IEEE Trans. Neural Network 14 236–242
[13] Maeda Y, Wakamura 2005 M Simultaneous perturbation learning rule for recurrent neural networks and its FPGA implementation IEEE Trans. Neural Network 16 6 1664 – 1672
[14] Rafael G, Ricardo C, Joaquin C, Angel C, Maeda, Wakamura M 2005 FPGA Implementation of a Pipelined On-Line Backpropagation J VLSI Signal Process 4 2
[15] Daniel F, Ramiro G, Roberto F, Julio P, Rafael C 2004 NeuroFPGA Implementing Artificial Neural Networks on Programmable Logic Devices”. Conf. on Design, automation and test in Europe 3
[16] Coric, Latinovic S, Pavasovic I, A 2000 FPGA Implementation Neural Networks Proc. Neural Network Applications in Electrical Engineering p 117 – 120

[17] Maeda Y, Wakamura M 2005 Simultaneous perturbation learning rule for recurrent neural networks and its FPGA implementation IEEE Trans. Neural Network 16 6 1664 – 1672.

[18] Goodman J W 1968 Introduction to Fourier optics McGraw Hill