Comparisson of different Neural Networks Architectures for Arm Gesture Recognition

Daniel Fernando Tello Gamarra
Electrical Energy Processing
Department (DPEE)
Universidade Federal de Santa Maria
Santa Maria - RS - Brazil

Anselmo Rafael Cukla
Electrical Energy Processing
Department (DPEE)
Universidade Federal de Santa Maria
Santa Maria - RS - Brazil

Solon Bevilacqua
Production Engineering Department
Universidade Federal de Goiás
Goiânia - GO - Brazil

Guillermo Alejandro Mudry
Mechatronic Engineering Department
Universidad Nacional de Misiones
Oberá, Misiones - Argentina

***Abstract*—The microsoft kinect sensor has open new doors in the field of human machine interaction, one of the features of kinect sensor is its ability of recognizing human skeleton joints. This work explores the use of the recognition human gestures using neural networks. Three different neural networks architectures will be tested in the paper, backpropagation, Radial Basis Function and ANFIS neural networks, experiments and results are shown in order to validate the proposed system for gesture recognition.**

I. Introduction

Technology is improving the way in which humans can interact with machines, an example of this optimization is the microsoft kinect sensor, originally created to be used in videos games, kinect sensor potential for other applications is in development, bringing the possibility of employing the microsoft kinect sensor for a plethora of diverse areas such as commerce, medicine, sales, education, industry, physiotherapy and robotics.

An important part of human machine interaction is gesture recognition, the microsoft kinect offers the possibility of identifying the human body skeleton. The human body skeleton has different joints and in order to identify gestures that are a sequence of movements is necessary to create a model that relates gestures and movements. Different machine learning algorithms have been proposed to build the referred model.

Some recent works have explored the use of Kinect in motion pattern identification. Rimkus in [1] proposes a systems based on a Kinect for hand motion recognition using feedforward neural networks. Sorce in [2] proposed a method based on a Kinect device and neural networks to identify whether a hand is open or close. Zhang in [3] uses a Kinect system based on Hidden Markov Models and Neuro-Fuzzy systems to model the movements adopted by players in Golf. Heickal in [4] uses the Kinect sensor to recognize full body skeleton used by the umpires in a cricket match, backpropagation and Naïve Bayes Classifier are used for gesture recognition. Also some new architectures using deep learning have been applied to the problem as in the works of [12], [13], [15] , [?]

This work proposes a system based in neural networks for gesture recognition, the gesture acquisition will be done with a Kinect sensor that will capture the skeleton data, so Multilayer Perceptron, Radial Basis Function and ANFIS neural networks will be employed for the gesture recognition. Besides, the application of these three different neural network architectures for gesture recognition, a comparison of the performance of the neural networks is analyzed. The paper is divided in 6 sections, after a brief introduction in the first section; the second section explains the theoretical background used in the paper; the third section describes the experimental setup; section fourth details the data storage phase and design of the neural networks; section fifth summarizes the results of the system and finally the last section elucidates the main conclusions of the paper and future works.

II. Theoretical Background

Neural networks have been employed in different ways such as regression, system identification and classification. It will be reviewed the principles and structure of the neural networks that have been used in our experiments.

A. Multilayer Perceptron (MLP)

Exist different kinds of neural networks and their classification is related to their learning algorithms or architecture, a multi-layer perceptron neural network (MLP) will be used, MLP neural networks are classified as networks with a supervised learning algorithm. The MLP has an architecture based in 3 layers, an input layer, a hidden layer and an output layer. In [5] is stated that networks with just two layers of weights are capable of approximating any continuous functional mapping. Based on the work written by Terra in [6] will be described a MLP neural network, for one hidden layer, presenting in the sample n, the input vector $x_n = [x_{1n}, x_{2n}, \ldots, x_{pn}]^T$, for a p-dimensional input vector and a q-dimensional output vector, the MLP output of the neuron k (where $k = 1, 2, \ldots, q$) is given by:

$$\hat{y}_k = \varphi_k(\Sigma w_{kj} \varphi_j(\Sigma w_{ji} x_{in})) \tag{1}$$

Where m is the number of neurons in the hidden layer, w_{kj} is the weight between the neuron j (hidden layer) and the neuron k (output layer), w_{ji} is the weight between the neuron i (input layer) and the neuron j (hidden layer), φ_k is the nonlinear activation function in the output layer and φ_j is the nonlinear activation function in the hidden layer.

B. Adaptive Neuro-Fuzzy Inference System (ANFIS)

As mentioned in [7], an ANFIS is a kind of adaptive network that acts as a framework for adaptive fuzzy inference systems. A fuzzy inference system is a popular computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning. The ANFIS neural network will be reviewed here briefly based in [7].

For a first-order Sugeno fuzzy model, a typical rule set with two fuzzy if-then rules can be expressed as:

$$\begin{array}{lllllllll} Rule & 1: & if & x & is & A_1 & and & y & is \quad B_1 \\ Then & f_1 & = & p_1x & + & q_1y & + & r_1 & \end{array}$$

$$\begin{array}{lllllllll} Rule & 2: & if & x & is & A_2 & and & y & is \quad B_2 \\ Then & f_2 & = & p_2x & + & q_2y & + & r_2 & \end{array}$$

In the structure of a neural network, nodes of the same layer have similar functions, we will denote the output node i in layer l as $O_{i,j}$. Where x and y are referred to any input to the network, and can be any kind of data, they could be image features or atmospherical information, depends of the network application. Every node i in this layer is an adaptive node with a node output defined by:

$$\begin{array}{lllll} O_{1,i} & = & \mu_{A_i}(x), & for \quad i = 1,2, & or \\ O_{1,j} & = & \mu_{B_{i-2}}(y), & for \quad i = 3,4, & \end{array} \tag{2}$$

Where x (or y) is the input to the node, and A_i or B_{i-2} is a fuzzy set associated with this node. A_i and B_i can be any parameterized membership function. For example, A_i can be characterized by the generalized bell function:

$$\mu_{A_i}(x) = \frac{1}{1 + [(\frac{x-c_i}{a_i})^2]^{b_i}} \tag{3}$$

Where a_i, b_i, c_i are the bell function parameters. Training the network consists of finding suitable parameters for the layers, gradient descendent methods and least squares methods are used in the training phase.

C. Radial Basis Function (RBF)

These kind of neural networks compute the activation of a hidden unit calculating the distance between the input vector and a prototype vector as stated in [5], the training of these networks is faster than the multilayer perceptron, these networks have a two stage training procedure, in the first stage are calculated the parameters of the radial basis function that are related to the hidden neurons, this first stage uses an unsupervised training technique, in the second stage are calculated the final weights of the layer. The radial basis function approach introduces a set of N basis functions, one for each point, that takes the form $\phi(\|x - x^n\|)$ where ϕ is some non-linear function, and depends of the Euclidean distance between x and x^n, where x and x_n are input vectors, the output of the mapping is then represented as a linear combination of the basis function:

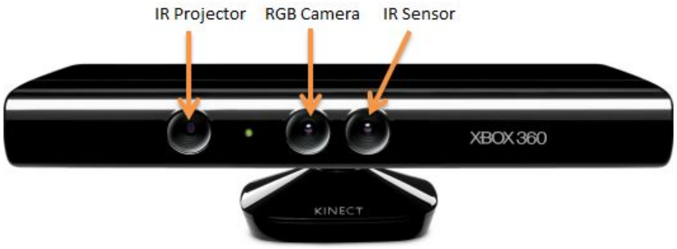

Fig. 1. Kinect Sensor (image taken from [9]).

$$h(x_n) = \sum w \sum w_n \phi(\| x - x^n \|) \tag{4}$$

and its matrix form is represented as:

$$\mathbf{\Phi w = t} \tag{5}$$

t, represents the network targets, if exists the inverse matrix, we can obtain:

$$\mathbf{w = \Phi^{-1} t} \tag{6}$$

There are many kinds of basis functions but the most common is the Gaussian:

$$\phi = exp(-\frac{x^2}{2\sigma^2}) \tag{7}$$

III. Experimental Setup

The kinect sensor will be employed in the paper, the kinect sensor is manufactured by the microsoft, the sensor was projected initially as a complementary device for the xbox videogames platform, the kinect sensor was created for the Jew company primesense, the kinect sensor has a camera and a sensor that is emitter and receiver of infrared rays and a microphones array, these different sensors generate the possibility of working with RGB digital images and the images depth information as is stated in [8]. Figure 1, shows the microsoft kinect sensor, that is the hardware equipment used for the paper.

Processing is a flexible software sketchbook [10], a program written in Java and using this software platform is employed to capture the human skeleton, the program will let us capture the skeleton data and store it. It was used as an initial program the code developed by Bryan Chung in [11], this program was modified in order to store the data that is of our interest and adapt it to our experiments this program uses processing and the library kinect4WinSDK. In Figure 2 it is possible to observe 4 subfigures obtained with the kinect, the first subfigure corresponds to the skeleton image, the second subfigure is the RGB image, the third subfigure of the left above is a segmented image of the person, the fourth subfigure is the depth image. The Matlab toolbox of neural networks will be used for the experiments of training and testing the captured data.

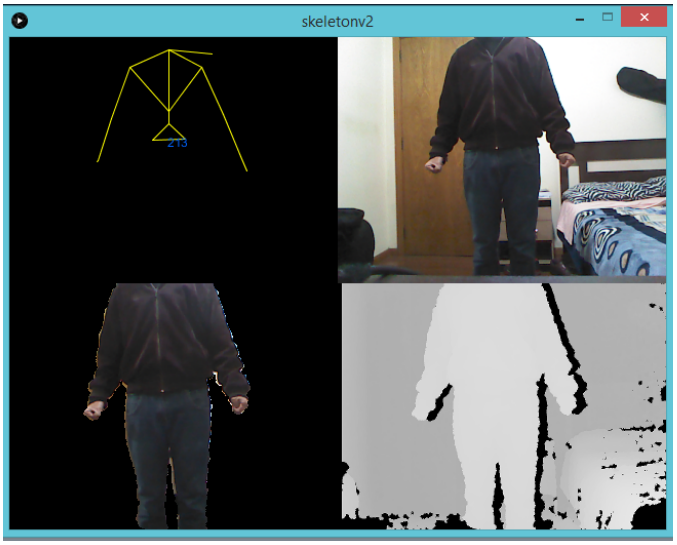

Fig. 2. Skeleton image, RGB and RGBD images capture with the microsoft Kinect sensor.

IV. Gesture Data Collection and Neural Network Design

The RGBD images taken from Kinect were captured using the program developed in the software processing, the joint data captured from the skeleton was stored. The joint positions chosen for the network input are the elbow, wrist and shoulder positions in coordinates x,y and z. The data captured in the experiments was integrated by 3840 samples. The data was divided in testing and training data, 2560 samples were chosen as training data and 1280 samples were chosen as testing data.

Afterwards, The Matlab toolbox of neural networks will be used for the experiments of training and testing the captured data. The training and sample data was normalized. The multilayer perceptron (MLP) neural network used 10 hidden neurons, the neural network will have 9 inputs corresponding to the wrist, elbow and shoulder positions in the coordinates x, y and z and one output that will be the gesture generated. The ANFIS neural network also used the same inputs and outputs for its architecture construction, and it constructed its membership functions with Gaussians. Table 1 shows the ANFIS neural network parameters constructed for the experiments.

TABLE I
ANFIS Neural Network Parameters

Parameter	Value
Number of nodes	92
Number of linear parameters	40
Number of nonlinear parameters	72

The radial basis function (RBF) neural network will employ the same number of inputs (9 inputs) based on the arm joints and one output based on the kind of gesture executed by the person. Six different gestures were captured with four different human subjects of different ages; the subject ages ranges were between 9 to 74 years old. Figure 3 shows the different gestures executed by the subjects, the gestures are labeled as $K1$, $K2$, $K3$, $K4$, $K5$ and $K6$ and encrypt linear and circular trajectories as shown in the figure.

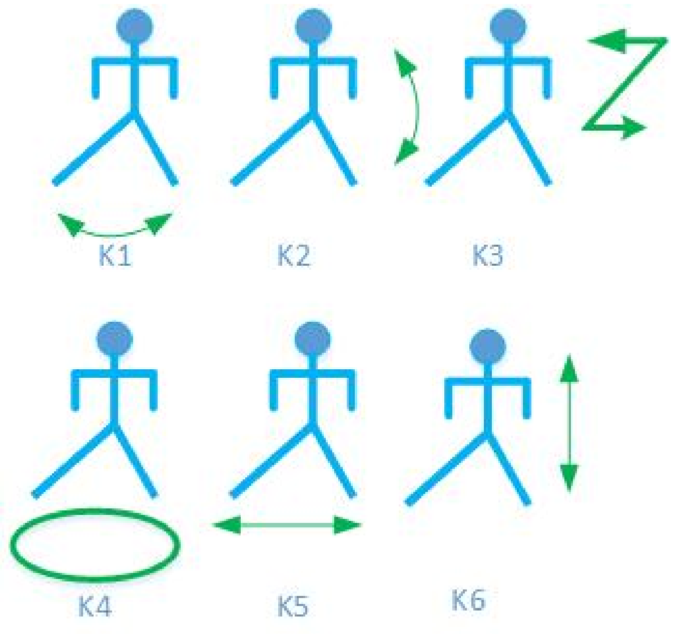

Fig. 3. Arm gestures trajectories .

Gestures Defined: $K1$ Gesture: Waving the arm in a horizontal plane. $K2$ Gesture: Waving the hand in a vertical plane. $K3$ Gesture: Drawing the letter ”Z”. $K4$ Gesture: Moving in a circular trajectory. $K5$ Gesture: Following a vertical line trajectory. $K6$ Gesture: Following a horizontal line trajectory.

V. Experimental Results

The results of the experiments with the neural networks are shown in this section. Figure 4 shows the multilayer perceptron neural network mean square error, it is possible to observe that the error is very low and the system will reach a solution in less than 12 epochs, the error will converge to less than 0.0001. The radial basis function neural network converges to a mean squared error of 0.00005 using 50 neurons, with a spread constant of 0.2 as it can be observed in figure 5.

The same experiments and data were used with the ANFIS neural network, Figure 6 shows that also the ANFIS neural network will converge to a solution after a finite number of epochs. Table number 2 resumes the different mean squared errors obtained with the neural network architectures used in the paper. Also table 3 shows the number of epochs employed for every neural network to converge to a determined error.

The results shown in the experiments indicate that the three different neural architectures can be used to the task of gesture recognition with the kinect sensor. However, the networks have different performances. If the mean squared error is analysed , the network with the minor mean squared error

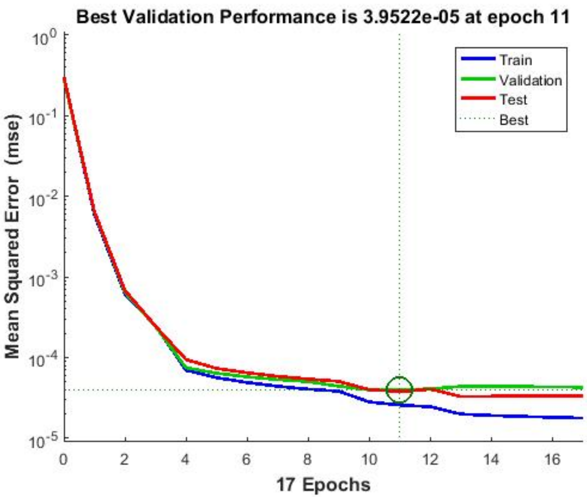

Fig. 4. Multilayer perceptron mean squared error.

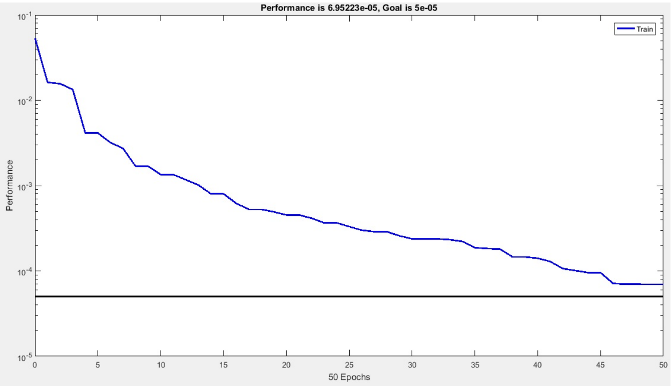

Fig. 5. Radial basis function mean square error.

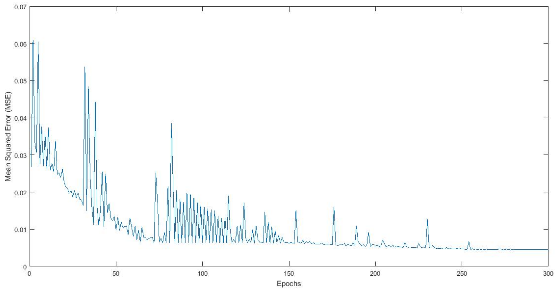

Fig. 6. Anfis neural network error.

TABLE II
NEURAL NETWORK-MEAN SQUARE ERROR

Neural Network	MSE
Multilayer Perceptron	0.000039522
RBF	0.0000695
ANFIS	0.00455096

TABLE III
NEURAL NETWORK-EPOCHS NUMBER

Neural Network	Epochs
Multilayer Perceptron	11
RBF	50
ANFIS	300

is the Multilayer Perceptron, followed by the Radial Basis Function and the ANFIS network. If the focus of the analysis is the number of epochs to reach a determine error, the Multilayer Perceptron uses just 11 epochs, followed by the Radial Basis Function with 50 epochs and the ANFIS neural network with 300 epochs.

Based on these results, we could sumarize that the Multilayer Perceptron is the one the shows the best performance, followed by the Radial Basis Function network in second place, and the ANFIS network in third place. Also the Multilayer Perceptron and Radial Basis Function networks have a simpler architecture compared to the ANFIS network that shows a more complex architecture and more number of parameters.

VI. CONCLUSIONS

This work shows the use of three different neural architectures for the task of gesture recognition, the data was capture using the microsoft kinect sensor, the skeleton data was identified and the skeleton joint data was extracted from the RGBD images generated with the Kinect sensor. Three different gestures of a person were used and the neural networks were trained. Results shown that both the Multilayer Perceptron, Radial Basis Function and ANFIS neural networks have a good performance for the task of gesture recognition in that order. But, the Multilayer Perceptron shows a better performance. The next step will be to use the gesture identification system developed in this work for the control of a mobile robot, that will recognize the gestures generated by a human and will move forward, backward or follow a predefined trajectory according to the hand movements.

REFERENCES

[1] K. Rimkus and A. Bukis and A. Lipnickas and S. Sinkevicius, *3D Human Hand Motion Recognition*, In 6th International Conference on Human System Interactions(HSI), pp. 180-183. IEEE Press, Sopot, Poland, 2013.

[2] S. Sorce and V. Gentile and A. Gentile, *Real-time Hand Pose Recognition Based on Neural Network Using Microsoft Kinect*, In 8th International Conference on Broadband, Wireless Computing, Communication and Application, pp. 344-350. IEEE Press, Compiegne, France, 2013.

[3] L. Zhang and J. Hsieh and J. Wang, *A Kinect-Based Golf Swing Classification System Using HMM and Neurofuzzy*, In 8th International Conference on Computer Science and Information Processing (CSIP), pp. 1163-1166. IEEE Press, Shaanxi, china, 2012.

[4] H. Heickal and T. Zhang and M. Hasanuzzaman, *Real-Time 3D Full Body Motion Gesture Recognition*, In IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 798-803. IEEE Press, Shenzhen, china, 2013.

[5] C. Bishop, *Neural Networks for Pattern REcognition*, Evolutionary Computation, 2012.

[6] M.H. Terra and R. Tinos, *Fault detection and isolation of robotic systems using a multilayer perceptron and a radial basis function*, In Proc. of the IEEE Conference on Systems, Man and Cybernetics, pp. 1880-1885. IEEE Press, San Diego, California, USA, 1998.

[7] J.S.R. Jang and C.T. Sun, *Neuro-Fuzzy Modeling and Control*, In Proc. of the IEEE , vol. 83, pp. 378–406, 1995.

[8] L. Cruz and D. Lucio and L. Velho, *Kinect and RGBD Images: Challenges and Applications*, In: Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T) , pp. 36–49, Ouro Preto, Brazil, 2012.

[9] Smoothing Kinect Depth Frames in REal-Time. http://www.codeproject.com/Articles/317974/KinectDepthSmoothing

[10] Processing.https://www.processing.org

[11] Bryan Chung https://www.magicandlove/blog/research/kinect-for-processing-library/

[12] M. Pfitscher and D. Welfer and M.A.D.L. Cuadros and D.F.T. Gamarra, *Activity Gesture Recognition on Kinect Sensor Using Convolutional Neural Networks and FastDTW for the MSRC-12 Dataset*, In: Intelligent Systems Design and Applications (ISDA) , pp. 230–239, 2020.

[13] J.S. Peixoto and A.R. Cukla and M.A.D.L. Cuadros and D. Welfer and D.F.T. Gamarra, *Gesture Recognition using FastDTW and Deep Learning Methods in the MSRC-12 and the NTU RGB+D Databases*, In: IEEE Latin America Transactions , pp. 2189–2195, v. 20, 2022.

[14] J.S. Peixoto and A.R. Cukla and M.A.D.L. Cuadros and D. Welfer and D.F.T. Gamarra, *Comparison of Different Processing Methods of Joint Coordinates Features for Gesture Recognition with a CNN in the MSRC-12 Database*, In: Intelligent Systems Design and Applications (ISDA) , pp. 590–599, 2021.

[15] J.S. Peixoto and A.R. Cukla and D. Welfer and D.F.T. Gamarra, *Comparison of Different Processing Methods of Joint Coordinates Features for Gesture Recognition with a RNN in the MSRC-12*, In: Intelligent Systems Design and Applications (ISDA) , pp. 498–507, 2022.