International Journal of Engineering & Technology, 7 (3.13) (2018) 1-4

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Artificial Neural Network Optimization with Levenberg–Maruardt Algorithm for Dynamic Gesture Recognition

Stephen John Dy[1], Matthew Adrianne Gonzales[2*], Lenard Lozano[3], Miguel Angelo Suniga[4], and Alexander Abad[5]

[1,2,3,4]*De La Salle University – Laguna Campus, Biñan, Laguna. Philippines*
[5]*Electronics and Communications Eng'g Dept., Gokongwei College of Engineering, De La Salle University, Manila, Philippines*
**Corresponding author E-mail: matthew_adrianne_gonzales@dlsu.edu.ph*

Abstract

Movement has long been a mode of expression and communication. A challenge arises when we try to bestow the ability to learn and recognize movements to machines, specifically computers, but with the development of sensor technology and the growing interest in machine learning algorithms, there is an opportunity to explore and formulate new approaches. The study focuses on the use of the Levenberg Marquardt Algorithm as an optimization algorithm for a multilayer Artificial Neural Network in constructing a predictive model for dynamic gestures. Extraction of the data set was made integral to the research. The study concludes that the network architecture is adequate for gesture recognition, with an average recognition rate of 83%, but a larger data set may show to improve this value.

Keywords: *Gesture Recognition; Industrial Robots; Neural Networks; Optimization; Robotics*

1. Introduction

The development of sensor technology in recent years can be attributed to the interest in vision-based technology applications in numerous scientific fields. Pattern recognition is especially significant in the research and development of more intuitive Human-Machine Interfaces (HMI), as current intermediaries are dated and do not fully exploit the capability of motion of the human hand and other body parts. Humans can perform and recognize gestures as one of many means of communicating with other humans with static and dynamic gestures. Static gestures are performed by creating a stationary pose or position, generally with one's hands. For static gestures, parameters such as the spatial relationships of the appendages or fingers on the hand allow the observer to identify the gesture made. Dynamic gestures are created with motion and may or may not exhibit recurrent patterns. These gestures, like static gestures, are usually understood naturally by humans, however an intimidating challenge is posed when we attempt to allow communication with machines, specifically computers, with gestures. The challenge of giving a machine the capability to recognize movements is one that has been tackled with predominantly with the use of machine learning algorithms. Consequently, much of the foundation for machine gesture recognition explores the use of Artificial Neural Networks (ANN).

The research explores the use of Levenberg–Marquardt algorithm in optimizing a Deep Feed Forward Neural Network through backpropagation to recognize and classify four gestures. A small-scale dataset will be used be used for training, validation, and testing of the neural network, with the objective of the study being to assess the feasibility of the network model in performing gesture recognition given a small-scale dataset. The input data is pre-processed with dimensionality reduction techniques to accommodate the limitations of the neural network.

2. Literature Survey

Creating a natural and fluid interaction between human and computer has long been the aim that hand gesture recognition system hopes to bridge. These recognized gestures can be used for many purposes, such as controlling a robot or conveying meaningful information [1]. How to form the resulted hand gestures to be understood and well interpreted by the computer is widely considered as the problem of gesture interaction [2]. The pattern recognition field has been studied thoroughly with different techniques and approaches. Human Interaction Devices (HID) have paved the way for more interactive and innovative ideas with regards to pattern recognition [3].

Reference [4] describes the different types of gesture recognition technologies. The authors identified these technologies as the Contact type, Device gesture technologies, and Vision-based technologies. **Contact type technologies** sense physical contact on a conventional touch pad or touch screen. **Device gesture technologies** use position tracking devices whose movements send signals that the system uses to identify the gesture, while **Vision-based technologies** use computer vision to track movements and extract raw data for posture and gesture recognition. In many cases, users do not want to wear tracking devices and computer-bound gloves since they can restrict freedom of movement and take considerably longer to set up than traditional interaction methods [4]. This disadvantage of wearable devices made vision-based technologies a better option for this study.

The first step in performing hand gesture recognition in literature is the **Segmentation process**. It involves dividing the input image into sections that are separated by boundaries [5]. The segmentation process depends on the type of gesture, a dynamic gesture requires the hand gesture to be located and tracked [5], while a static gesture, the input image must be segmented only once. Hand tracking can be done in two methods – either the video is divided

into frames and each frame have to be processed alone, wherein the hand frame is treated as a posture and is segmented in this case [5], or using some tracking information such as shape and skin color using some tools such as Kalman filter [5]. The next step is **Features Extraction**. This step is crucial to the success of the recognition process [6]. The different methods used different ways to identify the gesture, such as the hand contour and silhouette [6] or the position of the fingertips, the center of the palm, etc. [5]. These two steps fall under the category of data preprocessing.
After modeling and analysis of the input hand image, **gesture classification method** is used to recognize the gesture. Recognition process affected with the proper selection of features parameters and suitable classification algorithm [7]. Statistical tools used for gesture classification, Hidden Markov Model (HMM) tool has shown its ability to recognize dynamic gestures, besides, Finite State Machine (FSM), Learning Vector Quantization, and Principal Component Analysis (PCA). Neural networks have been widely applied in the area of hand gesture recognition [3].

3. Dataset and Preprocessing

The Leap Motion and Leap Motion API provided a means to utilize the sensor in extracting gesture data. The Leap Motion is an infrared camera capable of tracking the position and velocity of the user's palm, among other things. A program was made with the API by the researchers to record these movements. A set of five (5) people were asked to perform four distinct gestures using the device, repeated in succession twenty (20) times each. These gestures are: a heart, a diamond, a square, and a triangle. The participants were asked to initiate the recording by clenching their fist, moving their hand across the area of operation of the sensor and release once they had performed the gesture. The device records the x, y, and z coordinates of the velocity and position of their hand as they are performing the gesture.

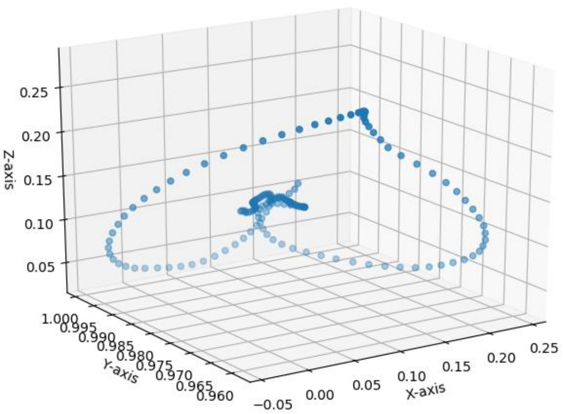

Fig. 1: Recorded heart-shaped gesture in a scatter plot

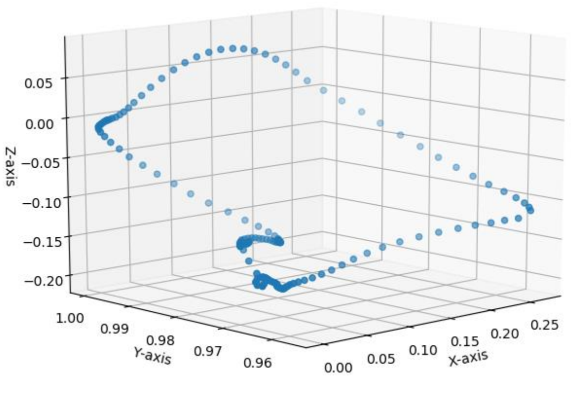

Fig. 2: Recorded diamond-shaped gesture in a scatter plot

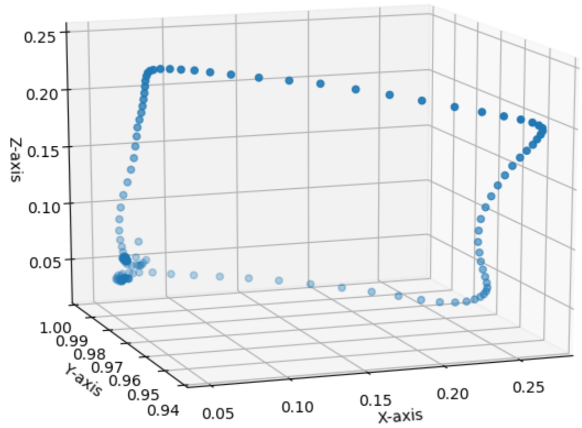

Fig. 3: Recorded square-shaped gesture in a scatter plot

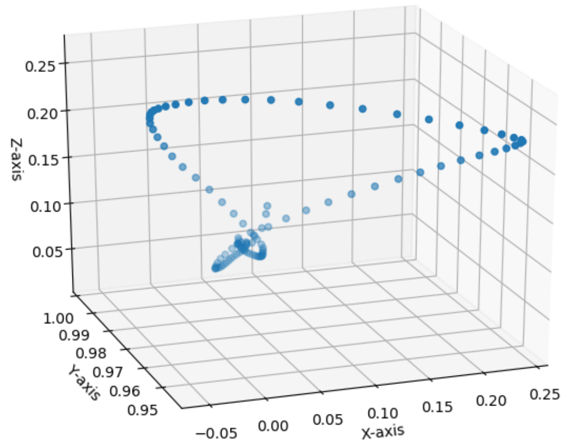

Fig. 4: Recorded triangle-shaped gesture in a scatter plot

Preprocessing of the dataset was performed using Principal Component Analysis (PCA) to emphasize the variations of the dataset. This method permits patterns in the data to surface for datasets with high dimensions. Using the linearly uncorrelated variables or n principal components, where n < m, the number of dimensions in the dataset, and the associated variances (explained variances) of each principal component, the data is represented with fewer dimensions and parameters and can be used for such network model. This preconditioned data is fed to the ANN, along with parameters such as the mean and maximum value of each coordinate. The instance of the dataset for the diamond gesture is as shown on (1).

$$Diamond_1 = [x_{max}, y_{max}, z_{max}, x_{mean}, y_{mean}, z_{mean}, var_1, var_2, \ldots, var_n]. \quad (1)$$

4. System Architecture

Artificial Neural Networks (ANN) are essentially crude electronic models of the neural behavior of the brain. They typically consist of hundreds of units or nodes which process the data in a complex communication network. ANN's offer a process to characterize synthetic neurons to solve complex problems in the same manner a human brain would. Typically, ANN's are utilized for processes such as modelling and control, character recognition, image recognition, models for analysis and adjustment of network parameters to increase availability [8], and stock forecasting [9].
ANN's consist of several units which act as mini-calculation devices. They accept real-valued input from multiple other nodes and they produce a single real valued output. The structure of an ANN is as follows:

1. A set of input units which take in information to be passed through a network which in turn produces an output. This forms the input layer.
2. A set of hidden units which take values from the input layer. A weighted sum of the output from the input units forms the input to every hidden unit. This forms the hidden layer.
3. A set of output units which dictate the category is propagated through the network. This forms the output layer.

Because the single feedforward networks were limited to learning linearly separated patterns, nonlinear layers between the input and output layers were implemented to separate the data, coupled with enough training to model any function to a target precision. In the study, the neural network consists of one (1) input layer, five (5) hidden layers with a hyperbolic tangent activation function, and an output softmax function layer and was implemented using the NeuPy library for Miniconda distributions of Python.

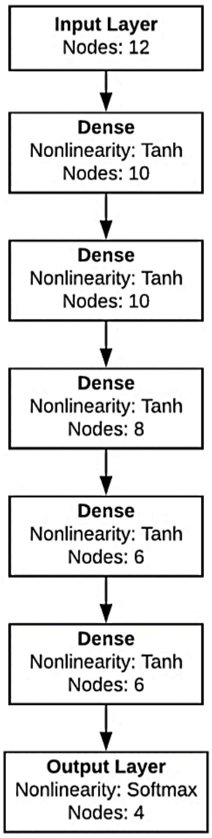

Fig. 5: Architecture of a Neural Network

This is what is known as a Multi-Layer Perceptron (MLP). These networks are often trained with the Back Propagation (BP) algorithm. The BP algorithm is supervised, which means it maps the process inputs to the desired outputs by minimizing the errors between the desired outputs and the calculated outputs driven from the inputs and network learning.

The Levenberg–Marquardt optimization algorithm is used to decrease the mean square error between the calculated and desired outputs of the MLP network. Assuming there are n process inputs, x, and m desired outputs, d, with the network with a sample size of N, there exists the value for process input x_{it} and desired output d_{jt}, where i is a positive integer less than or equal to n, j is a positive integer less than or equal to m, and t is a positive integer less than or equal to N. With that, the network outputs' mean square error is therefore the equation which is as shown on (2) [10].

$$E = 0.5 * \Sigma^{N}_{t=1}\Sigma^{m}_{j=1}(d_{jt} - y_{jt})^2. \quad (2)$$

The training set is fed to the neural network with the respective outputs for the adjustment of weights. The neural network initializes with random weights and the weights are adjusted through the relationship of the weights of the network and the mean square error. In the case of this study, the Levenberg–Marquardt algorithm is in each iteration of weight update to minimize the error. The Levenberg–Marquardt Algorithm can be described as a curve-fitting method and as a combination of two minimization algorithms: the gradient descent method and the Gauss-Newton method [11]. It is a minimization technique that locates the minimum of a multivariate function that is expressed as the sum of squares of non-linear real valued functions. The sum of the squared errors is reduced by moving toward the direction of steepest descent. At each iteration, the Levenberg–Marquardt Algorithm chooses either the gradient descent or GN and updates the solution.

Specifically, by adding an extra term that is proportional to the identity matrix I in the normal equation as obtained in the Gauss-Newton method as shown on (3),

$$(J^TJ)\Delta a = J^T(y - f(a)). \quad (3)$$

such equation was modified to as shown on (4)

$$(J^TJ + \lambda I)\Delta a = J^T(y - f(a)). \quad (4)$$

where J is the Jacobian Matrix. The iterative update is dependent on the algebraic parameter λ, a non-negative damping factor which smooths out the graph. If the value of λ is small, the update is Gauss-Newton. Otherwise, if the value of λ is large, the update is gradient descent [12]. The Gauss-Newton is faster and more accurate than the gradient descent when close to the minimum error. Initially, λ is set to a large number to set LMA in Gradient Descent. After a single iteration, the squares of the errors are then compared to a user-specified threshold to determine whether the step is accepted or not. If the resulting parameter is sufficiently better, it replaces its previous values and λ is reduced by a factor. Otherwise, λ is increased by a factor, and the algorithm proceeds to the next iteration. As the monitored parameter approaches its optimal value, LMA switches to Gauss-Newton method. Convergence between the monitored parameter and its optimal value is then achieved [13]. Based on the LM algorithm, every network connection weights are updated as shown on (5) [14].

$$w_{k+1} = w_k - (J^T_kJ_k + \mu I)^{-1}J_ke_k. \quad (5)$$

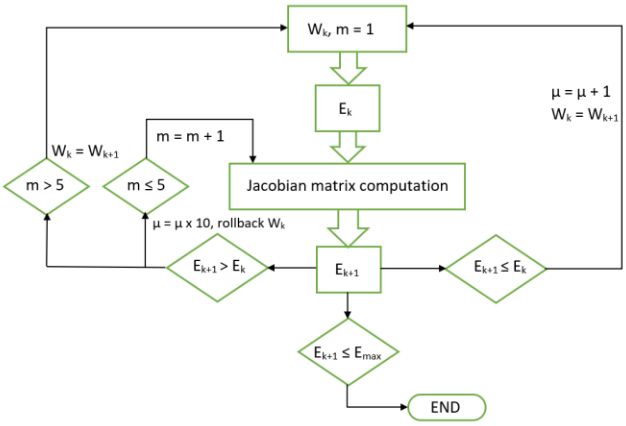

Fig. 6: Architecture of the Levenberg–Marquardt Algorithm

5. Experiments and Results

Following the extraction and preconditioning of the data, the data is shuffled and then divided into a training, validation, and testing set. A total of 400 recordings for the four (4) gestures are distributed as shown on

Table *1*.

Table 1: Distribution of data sets

	Recorded gestures
Training set	200 (50 for each gesture)
Testing set	100 (25 for each gesture)
Validation set	100 (25 for each gesture)
Cumulative data	400 (100 for each gesture)

Each instance in the data set is tied to a corresponding four-element list that indicates which gesture the recorded data represents. Each element of the list can take a value within the range(0,1). The threshold for a positively recognized gesture is 0.6 and because the output Softmax layer ensures that the sum of all elements is 1, it ensures that there are only two output states:

1. There is a single positively recognized gesture. (One element has a value greater than 0.6.)
2. There is a single positively recognized gesture. (One element has a value greater than 0.6.)

The network converges at an MSE of approximately 0.07 at less than 1000 epochs, with an update factor of 1.000015 and an μ of 1. The results of the testing set are as shown on Table 2.

Table 2: Results

	Number of gestures recognized (out of 25)	**% recognized**
Diamond	16/25	64%
Heart	22/25	88%
Square	22/25	88%
Triangle	23/25	92%

6. Analysis & Conclusion

The predictive model exhibits an 83% average recognition rate, with a minimum of 64% and a maximum of 92%, thus the model proves to be adequate in recognizing and classifying gestures.

Data sets obtained from libraries would likely provide a higher average recognition rate, but more stringent data gathering would yield the same effect. A larger data set may be more suitable to provide a greater recognition rate, and careful scrutiny of the per-epoch analysis may point to errors contributed instances in the data set. Furthermore, modular or mixed neural networks may be able to exploit more verbose characteristics of the data concurrently.

Acknowledgement

The authors would like to thank Almighty God for making everything possible according to His divine will and providence, especially in giving them strength, knowledge, and wisdom towards the completion of this document. They would also like to thank their family and friends, De La Salle University – Manila (DLSU), De La Salle University – Laguna Campus, and Department of Science and Technology – Engineering Research and Development for Technology (DOST-ERDT) for funding and helping us to finish this study.

References

[1] G. R. S. Murthy and R. S. Jadon, "A Review of Vision Based Hand Gestures Recognition," International Journal of Information Technology and Knowledge Management, vol. 2, no. 2, pp. 405-410, 2009.

[2] N. A. P. Garg and S. Sofat, "Vision Based Hand Gesture Recognition," World Academy of Science, Engineering and Technology, vol. 49, pp. 972-977.

[3] B. M. Lee-Cosio, C. Delgado-Mata and J. Ibanez, "ANN for Gesture Recognition using Accelerometer Data," The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science, no. 3, pp. 109-120, 2012.

[4] S. Naik, A. HR, A. KN and B. SP, "A Study on Automotive Human Vehicle Interaction Using Gesture Recognition Technology," International Journal of Multidisciplinary in Cryptology and Information Security, vol. 1, no. 2, pp. 6-12, 2012.

[5] N. Ibraheem, M. Hasan, R. Khan and P. Mishra, "Comparative Study of Skin Color based Segmentation Techniques," 2012.

[6] X. Li, "Gesture Recognition Based on Fuzzy C-Means Clustering Algorithm," 2003.

[7] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," EEE Transactions on systems, Man and Cybernetics, Part C: Applications and reviews, vol. 37, no. 3, pp. 311-324, 2007.

[8] L. Villanueva and R. C. Gustilo, "Artificial neural network based antenna sensitivity assignments for chaotic internet service provider network architecture," International Journal of Engineering & Technology, vol. 7, no. 2.3, pp. 14-17, 2018.

[9] C. B. Bangal, "Automatic Generation Control of Interconnected Power Systems Using Artificial Neural NetworkTechniques," 2009.

[10] Y. Huang, "Advances in Artificial Neural Networks – Methodological," Algorithms, vol. 2, pp. 973-1007, 2009.

[11] H. P. Gavin, "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problem," 22 March 2017. [Online]. Available: http://people.duke.edu/~hpgavin/ce281/lm.pdf. [Accessed April 2018].

[12] R. Wang, "Levenberg-Marquardt algorithm," 12 February 2015. [Online]. Available: http://fourier.eng.hmc.edu/e176/lectures/NM/node37.html. [Accessed 28 April 2018].

[13] S. K. Yewale and P. K. Bharne, "Hand gesture recognition using different algorithms based on artificial neural network," in 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), Udaipur, India, 2011.

[14] H. Yu and B. M. Wilamowski, "Levenberg–Marquardt Training," in Industrial Electronics Handbook, vol. 5 – Intelligent Systems, 2nd Edition, chapter 12, CRC Press, 2010, pp. 12-1 to 12-15.