

Implementation of Neural Networks for Direction Control of DC Motor

Abdullah Masrub* , Ramadan Omer 

Department of Electrical & Computer Engineering, Faculty of Engineering, University of Elmergib, Alkhoms, Libya

Email. a.masrub@elmergib.edu.ly

Abstract

In this paper, a neural network was implemented to control the direction of a DC motor. In this context, attention was paid to the field of artificial intelligence related to neural networks. The network targeted for training is small in size, and the tasks required of it are limited and include the direction of rotation of the DC motor axis, which has two directions (clockwise rotation or counterclockwise rotation). DC motors are used in applications such as robotics and automated manufacturing. It is a rotary actuator or linear actuator that allows control of linear direction. The main purpose of this work is to provide a clear view of the importance of using neural networks in the control field. In this study, the ANN is trained by the back-propagation algorithm, and a multilayer neural network is composed to implement the controller that controls the operation of the model using the back-propagation learning algorithm that regulates the function of the DC motor system. It was noticed that changing some parameters such as weights, learning rates, number of hidden layers, or number of nodes can enhance the output and generate the proper control of the direction.

Keywords. Neural Network, DC-Motor, Back-Propagation Algorithm, Artificial Intelligence.

Introduction

Recently, considerable research efforts have been focused on using neural networks in control systems. The fast development of software and hardware in recent years is the reason behind that attention. Principally, Artificial Neural Networks (ANN) have regained attention for significantly improving the performance of many intelligent processing units and tasks. The neurons of the network are connected by links, and each link has a numerical weight associated with it. Weights are the basic means of long-term memory in ANNs. They express the strength of each neuron input. A neural network learns through repeated tuning of these weights. To build an ANN, it must be decided first how many neurons are to be used and how the neurons are to be connected to form a network. In other words, the network architecture must be chosen first, and then the learning algorithm must be used. Finally, the neural network is trained such that the training is done by initializing the weights of the network and updating the weights from a set of training examples [1-2].

Typically, the network consists of an input layer of several neurons, at least one middle or hidden layer of computational neurons, and an output layer of computational neurons. The input signals are propagated in a forward direction layer by layer, and each layer in a multilayer neural network has its own specific function. The input layer accepts signals from the outside environment and distributes them to all neurons in the hidden layer. However, the input layer rarely includes computing neurons. To complete the process, the output layer receives signals from the hidden layer and produces the network's final output. Neurons in the hidden layer detect the features, where the weights of the neurons represent the features hidden in the input patterns. These features are then used by the output layer, which has the role of determining the output pattern. In such systems, with one hidden layer, we can represent any continuous function of the input signals, and with two hidden layers, even discontinuous functions can be represented. As the received signals are processed, they might be sent to the computer centre via the PC-control board (Arduino board) [3-4]. Usually, the movement in the machine or motor can be controlled through one of the artificial intelligence algorithms by reading variables of the environment using sensors and then applying the reaction step.

In other words, the purpose of establishing control of an industrial machine is to provide real skills to control the equipment that cannot be controlled directly due to environmental conditions, place conditions, or the circumstances of the application process, such as accuracy, which provides the necessary technical control in machines without constant surveillance operations. The term back-propagation strictly refers only to the algorithm for computing the gradient. It is often used to refer to the entire learning algorithm, including how the gradient is used [5-6]. Some control techniques can solve different issues, such as DC-motor direction. A DC motor is a rotary actuator or linear actuator that allows precise control of direction. Many controllers have been proposed to control digital DC motors, including optimal control, sliding mode control, Adaptive control, fuzzy logic, etc. However, these controllers are complex; hence, it is difficult to implement. The main problem in controlling a DC motor as a direction control drive is to achieve the right direction. Most direction controllers expose several problems because of the nonlinearities in the system.

Moreover, these nonlinearities are often unknown. Therefore, these system nonlinearities affect the performance of direction control of the DC motor system. For example, the commonly used Proportional-Integral-Derivative (PID) controller is simple and easy to implement, but this controller suffers from poor performance if it has nonlinearities and uncertainty issues. The PID controller is also incapable of working well for non-linear systems, and particularly complex and ambiguous systems that has no precise mathematical models [7-10]. The reason for selecting this study is the lack of practical applications of the idea of ANN, trying to find out their problems, and then trying to find solutions for them. To overcome these difficulties, the neural network is developed to ensure that the DC motor is operable in any circumstance of situations.

Methods

The proposed system has been developed in order to create an intelligent control system to solve the problems of manufacturers in connection with equipment that is difficult to control manually. This system works by receiving signals from sensors and sending them to the computer centre via the PC-control board (Arduino board). Then the processed signals are used to control the response of the device that will be controlled. Figure 1 shows the mechanism of the system and the time stages of implementing the system through different stages. The relationship between the computer system and the engine is an (on/off) relationship. Whereas the relationship between the neural network and the motor is the process of starting the DC motor, turning off the DC motor, and determining the direction of rotation of the motor. The role of the sensing system is to receive inputs from the environment, which have a value of zero or one for representation. The next stage starts when these inputs enter the neural network for processing to calculate the output. As a result, the outputs should activate the DC motor by rotating it clockwise or counterclockwise. In addition, the sensors can determine whether the system turns off the motor or does not rotate it at all.

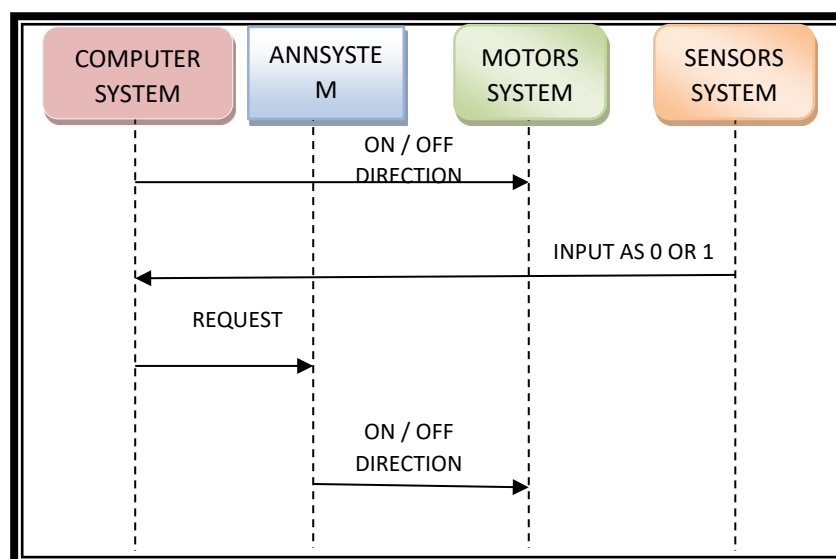


Figure 1. Sequence diagram of the system.

System Inputs

As the inputs to the system are represented by sensors, many types of sensors can be implemented. Some types are the distance sensor, temperature sensor, humidity sensor, and touch sensor. These sensors can be chosen according to the nature of the system's work, or according to the user's need. In this work, the strategy is based on three inputs, where each input represents a sensor. The nature of the inputs that the system deals with is in binary form, where the number 0 represents the absence of a signal from the sensor, and the number 1 represents the presence of a signal *Figure 1. Sequence diagram of the system*. Signal from the sensor. Hence, this representation of the binary system will design the input matrix. The input matrix consists of all the possibilities that may enter the system. According to the binary system, where the inputs represent binary numbers zero or one, the following (Table 1) shows all possibilities.

Table 1. The input matrix.

Vector 1	Vector 2	Vector 3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

System Outputs

In neural networks, generally, there are two types of outputs. The first type represents the outputs to be accessed, which represent a typical state of the system. The other type represents the outputs that the neural network calculates. However, three outputs were designed in this system, and each output represents a state related to the engine's response. Firstly, rotation of the motor axis clockwise; secondly, rotation of the motor axis counterclockwise; and finally, the state is the motor not making any response. The designed outputs can be represented in the form of a matrix or a table as shown in (Table 2).

Table 2. The output matrix.

Vector 1	Vector 2	Vector 3	Event
1	0	0	Right side rotate
0	1	0	Left side rotate
0	0	1	Stop

Data Processing

Generally, the neural network has been divided into three layers, and each layer represents a component of the network. The first layer represents the input layer, which consists of three neurons. The second layer is the hidden layer, which also consists of three neurons. In such a structure, the hidden layer has a role in connecting the input layer with the output layer. Each node inside the hidden layer performs certain operations with the input layer and other operations with the output layer. The inputs and outputs of the hidden layer are responsible for updating the weights, which are related to training the network. The last layer is the output layer, which contains the output to be reached. The output layer in this system consists of three outputs and has importance in determining the reaction of the system. This can be done during the training process, where it is possible to determine the error between the designed output and the calculated output. The amount of this error results in adjusting the weights between the hidden layer and the output layer, and between the hidden layer and the input layer.

Decision Table

The decision table is a grid-based visual representation for specifying which actions to take depending on the given conditions. It can be considered as an intelligent system that generates a set of actions. Each decision corresponds to a certain variable or might predicate which possible action is listed among the condition alternatives. As shown in the system decision table (Table 3), the condition alternatives are in the form of Boolean values, and the action entries are in the form of check-marks, representing which of the actions in a given column are to be performed. In total, there are three conditions for setting the rules, organizing the decision table, and then setting the events, which are also three events related to the outputs of the network.

Table 3. Decision table for rules used to control the machine.

		Rules							
		1	2	3	4	5	6	7	8
Sensor1		Y	Y	Y	Y	N	N	N	N
Sensor2		Y	Y	N	N	Y	Y	N	N
Sensor3		Y	N	Y	N	Y	N	Y	N
Actions	Right Rotate Side	✓	✓	✓	✓				
	Left Rotate Side					✓	✓	✓	
	Stop								✓

Results and Discussion

To provide a concise and precise description of the experimental results, their interpretation that can be drawn in graphs, different comparisons are made with changing some parameters such as weights, motivation function, learning rates, number of hidden layers, or number of nodes. In this work, we focus on affecting the learning rate on the network performance. The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. Choosing the learning rate is challenging, as a value too small may result in a long training process that could get stuck, whereas a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process. The learning rate may be the most important hyperparameter when configuring the neural network. Therefore, it is vital to know how to investigate the effects of the learning rate on the model performance and to build an intuition about the dynamics of the learning rate on model behavior. The obtained results are discussed by making a comparison of many variables, including weights in the period of (0-1), and the learning rate is changed with different values in the period of (0-3), then errors are calculated.

Table 4. Errors with learning rate in period of (0 – 3).

Epoch Learning rate	errors in 100	errors in 500	errors in 1000	errors in 1500	errors in 2000
0.1	0.2993701	0.271311	0.227968	0.331263	0.252757
0.3	0.3276702	0.145868	8.58E-02	0.111988	0.120221
0.5	0.3068014	9.18E-02	5.14E-02	0.070902	7.80E-02
0.7	0.2796049	6.77E-02	0.0387987	5.45E-02	6.01E-02
0.9	0.2494063	5.40E-02	0.0322071	0.045373	5.00E-02

1.1	0.2181064	0.045583	2.81E-02	3.95E-02	0.043439
1.3	0.1886991	3.99E-02	2.52E-02	3.53E-02	3.87E-02
1.5	0.163527	0.03587	2.30E-02	3.22E-02	3.51E-02
1.7	0.143364	3.28E-02	2.13E-02	2.98E-02	0.032265
1.9	0.1278153	3.05E-02	1.99E-02	2.78E-02	2.99E-02
2.1	0.1161238	0.028542	0.0188138	2.61E-02	2.80E-02
2.3	0.107605	2.69E-02	1.79E-02	2.47E-02	2.63E-02
2.5	0.1016853	2.56E-02	1.70E-02	2.35E-02	2.49E-02
2.7	9.78E-02	2.44E-02	1.63E-02	2.25E-02	2.36E-02
2.9	9.53E-02	0.023419	1.57E-02	2.16E-02	2.25E-02

(Table 4) shows the errors with different learning rates in different epochs. We notice that when the learning rate increases, the errors decrease; that is clear among different learning rates. For more clarification, (Figure 2) shows the average errors with learning rates in period of (0 -3).

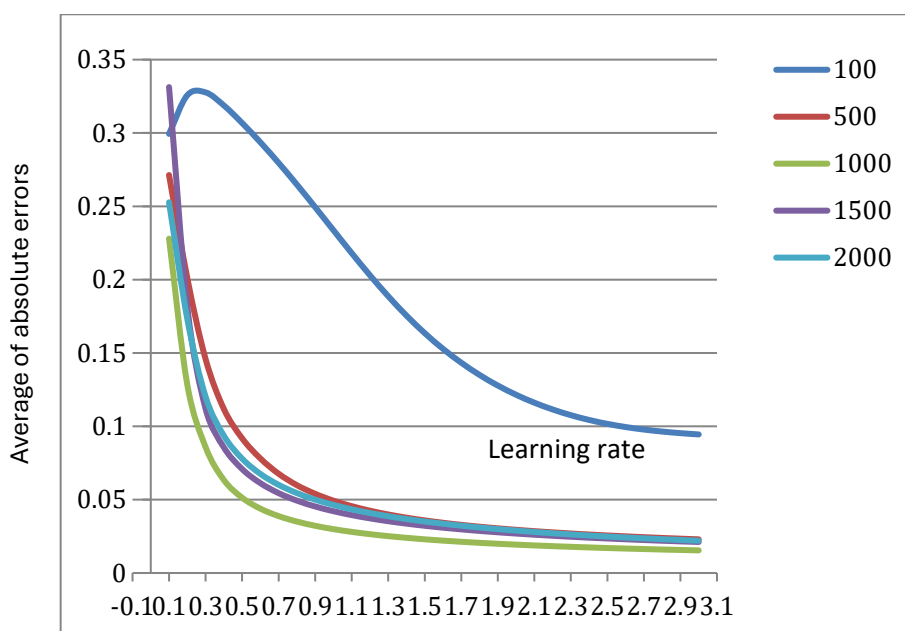


Figure 2. Average errors with learning rate in period of (0 – 3).

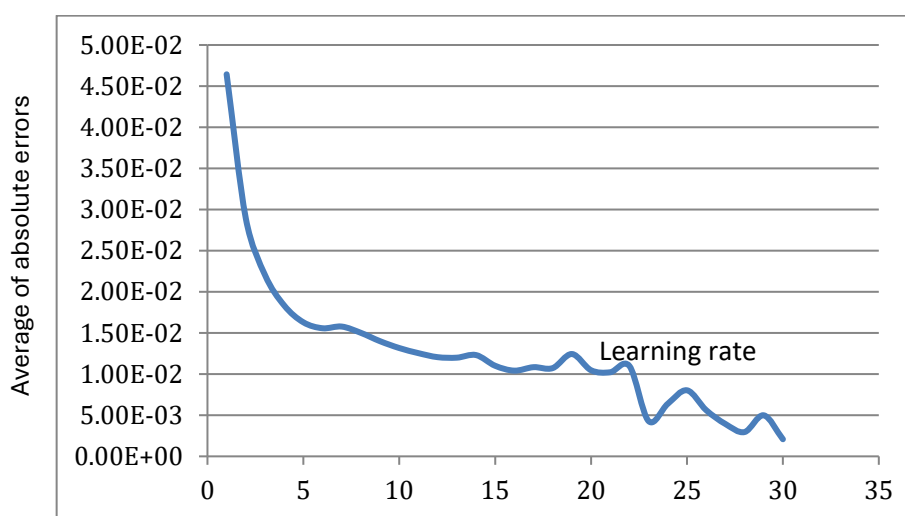


Figure 3. Average errors with learning rate in period of (0 – 30).

We continue our comparison with different learning rates in different epochs. Figure 3 shows the average errors with learning rates in period of (0 -30). From the previous figures, it can be noticed that whenever the learning rate value is high, the errors decrease, and net learning is faster. When the learning rate equals one, it can be used to simplify calculations. Moreover, the learning rate can be chosen randomly through practice.

Conclusion

The results indicate that increasing the learning rate generally reduces the network error and improves learning efficiency within the tested range. However, the optimal learning rate should be selected carefully to balance convergence speed and training stability. Based on the results presented in Table 4, a learning rate of 2.9 achieved the lowest error values among the tested rates, although further validation is needed to confirm its generalization performance.

Conflict of interest

The authors declare no conflict of interest.

References

1. Atri A, Ilyas M. Speed Control of DC Motor using Neural Network Configuration. *Int J Adv Res Comput Sci Softw Eng*. 2012 May;2(5).
2. Allu AA, Antar RK. Design and Modeling of Speed Sensorless Control of DC Motor Drive System. *Al-Rafidain Eng*. 2014 Dec;22(5).
3. Mishra M. Speed Control of Dc-Motor Using Novel Neural Network Configuration [master's thesis]. Rourkela (India): Department of Electrical Engineering, National Institute Technology; 2009.
4. Asha KR, Tasleem P. Real Time Speed Control of a DC motor by Temperature Variation using LabVIEW and Arduino. In: *International Conference on Recent Advances in Electronics and Communication Technology*; 2017; p. 16-17.
5. Teo CH. Backpropagation Neural Networks in Adaptive Control of Unknown Nonlinear Systems [master's thesis]. Monterey (CA): Naval Postgraduate School; 1991.
6. Bielawski L, Lewand R. *Intelligent Systems Design: Integrating Expert Systems, Hypermedia and Database Technologies*. New York: John Wiley & Sons Inc.; 1991.
7. Gavran M, Fruk M, Vujisić G. PI controller for DC motor speed realized with Arduino and Simulink. 2017; p. 22-26.
8. Boaz L, Priyatharshini S. DC Motor Direction and Speed Control by Arduino through RF Wireless Technique. *Int J Innov Res Comput Commun Eng*. 2016;4(2).
9. Obaid H, Sohail G, Muhammad NA, Amjad K. Speed and Direction Control of a DC Motor using Dual Converter Technique. *Int J Eng Works*. 2019;6(4):120-5.
10. Wadile KA, Chillarge SR, Jadhav DD. Speed Control and Direction Control of Dc Motor Using Arm7. *Int J Eng Comput Sci*. 2015 Feb;4(2):10524-8.