

Execution and analysis of classic neural network algorithms when they are implemented in embedded systems

Alfonso Jair Martínez López¹, José Raúl Pale Suarez¹, and David Tinoco Varela^{2*}

¹Facultad de Estudios Superiores Cuautitlán, UNAM, ITSE. México.

²Facultad de Estudios Superiores Cuautitlán, UNAM, Departamento de ingeniería, ITSE. México.

Abstract. Many algorithms related to neural networks are used in a large number of applications, most of them implemented on computational equipment that have great processing and storage capacities, however, new communication schemes such as the Internet of Things, need that neural algorithms can be executed from small electronic devices, devices that do not have large storage or processing capacities, but they can function as intelligent control centres for the different "things" connected to the Internet.

Currently, there are various electronic devices that allow generating low-cost intelligent technology projects that permit interaction within the Internet of things, such as the Arduino UNO, Tiva-C, and BeagleBone development boards. In this project, we present the analysis of the Perceptron, ADALINE and Hopfield neural network algorithms, when they are executed within the three mentioned development boards, in order to define the best tool to be utilized when using such neural schemes and few data are processed. Economic cost, temporary response and technical capabilities of electronic devices have been evaluated.

Keywords: Embedded Systems, BeagleBone, Tiva-C LaunchPad, Arduino, Neural Networks, Perceptron, ADALINE, Hopfield, Internet of Things.

1 Introduction

Nowadays, it is of great scientific and technological interest the development of engineering systems that can interact with their environment, obtain information from it, process such information intelligently, define an output behaviour, and send and receive data through the Internet, looking for these developments to be low cost so they can be accessible for any user.

Several development boards that can be found at low commercial cost, are used for the creation of various technological projects, such as the Raspberry Pi, Arduino (in its various forms), BeagleBone, Tiva-C, FPGA (field-programmable gate array), Jaguar One, Parallax Propeller, among others. One of the main characteristics of this type of boards is its small size, which is ideal for applications that require mobility. Additionally, through its input and output pins, projects developed with this kind of boards can interact with their environment in an easy and fast way. However, one of the main disadvantages of these embedded systems is that they do not have large processing neither storage capacities.

Currently, it is not only looking for development projects to be mobile, cheap, efficient, but also to make them smart. However, algorithms that process and analyse information intelligently, such as *Artificial Neural Networks* (ANN), usually require large processing capacities [1], which limits the use of the mentioned development boards. In some cases,

these boards are only used as a bridge between the data acquisition and more powerful computing equipment that processes such data [2], and in some other cases, the processing is divided between a development board and a data processing system of high capacity [3, 4].

This project seeks to analyse the algorithms of Perceptron, ADALINE (*Adaptive Neural Network*), and Hopfield neural network, when these are implemented in the Arduino UNO, Tiva-C LaunchPad and Beaglebone development boards, analysing their behaviour with respect to time, and the capacity to respond efficiently when there are not large amounts of data to be processed.

2 Preliminaries

2.1. Embedded systems

An embedded system is an electronic device considered to be of specific purpose, these devices have all the necessary elements to achieve their goals. They can be seen as small computers that have a processing unit, RAM and ROM memory elements, input and output ports, internet connection, and in the specific case of the embedded systems that we are going to work with, input and output pins that allow to obtain information from their environment through the control of sensors and different types of actuators.

* Corresponding author: dativa19@hotmail.com

Currently there are different types of embedded systems, among which we can find development boards based on FPGA, based on microcontroller and based on microprocessor; systems that may contain their own Operating System (OS) or depend on a development environment running from a PC. This type of systems can be programmed directly in assembly language, or by means of a high level language such as C / C++.

The development of this type of systems has allowed the design of a countless number of technological projects [5-7], as well as they have served as platforms to bring students of different levels closer to scientific and engineering design [8, 9].

Although there is a great variety of this type of platforms, in this study only 3 of them will be treated, Arduino UNO, Tiva-C LaunchPad and BeagleBone. In figure 1, you can see the three embedded systems presented in this project. In table 1, you can see the technical characteristics of each of these development cards.

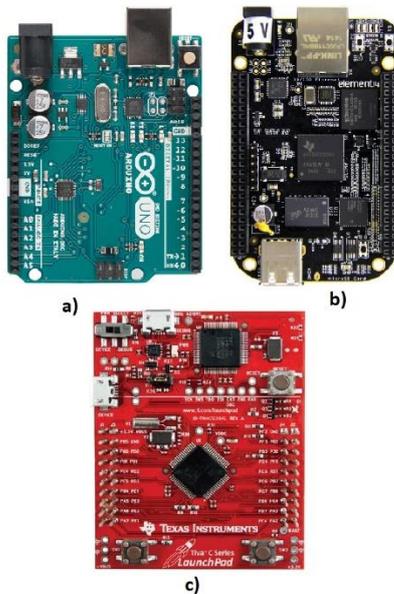


Fig. 1. Different development boards: a) Arduino UNO, b) BeagleBone, c) Tiva-C Launchpad.

2.1.1 Arduino UNO

The Arduino platform is perhaps the best known of all the existing development boards at this time. Arduino was developed at the *Interaction Design Institute Ivrea* in Ivrea, Italy. This project was born with the intention that beginners and professionals could develop technological projects in an easy and economical way. It is currently managed under the *open hardware* license. Among all the platforms used in this project, Arduino UNO is the development board with the most developed scientific and technological applications, some of them can be seen in [10], [11] and [12].

2.1.2 BeagleBone

BeagleBone is a board of low consumption, but of great computational power. Developed by *Texas Instruments* in association with *DigiKey* and *Newark element14*. This

board was designed to develop open source software, and like other boards of this type, it was also designed with the intention that it could be used by students to develop projects based on free software and hardware. It is currently marketed under the *share-alike* license of *Creative Commons*. It is a board that contains its own OS, which can be Debian, Android or Ubuntu. There are several versions of this development board, to visualize its characteristics and research areas of each of them, can be read [13].

BeagleBone has been used for the development of different technological projects, for example as a control system for mobile navigation of a differential unit using a single camera connected to the robot platform [14]. This board has also been the core of schemes related to the *Internet of Things* (IoT), in [15] an intelligent multisensor system was developed, where a multipurpose hierarchical network composed of intelligent nodes capable of obtaining and managing heterogeneous data and spaces is created according to the IoT paradigm. This board has also served for the development of pollution monitoring in a city, measuring the level of carbon dioxide and carbon monoxide in the air along with the location of the Global Positioning System (GPS) as can be seen in [16], among other projects [17, 18].

2.1.3 Tiva-C LaunchPad

This development platform focuses on the design of electronic prototypes, is based on the Texas Instruments microcontrollers family. It uses the M4C123G microcontroller. It can be programmed through the C language, or through the *Energia* project, which is an *Open Source* variant of the integrated development environment (IDE) of Arduino. The characteristics and manner of use it can be seen in [19].

Tiva-C has been used for projects such as the development of a multispectral information acquisition system to analyse vegetation cover, where the development board is coupled with an unmanned aerial vehicle, which allows the manoeuvrability of latitude and longitude, as can be seen in [20]. It has also been used for the construction of a laboratory prototype of a wireless sensor network (WSN) [21], among other projects [22, 23].

2.2 Neural networks

ANNs are computational schemes that try to imitate the way in which the human brain learns. These schemes have become very popular tools for solving different types of problems, mainly recognition and search of patterns, as well as imaging processing. We can see some applications and different schemes in [24] and [25].

Although this type of schemes has mostly been developed in projects dependent on high-performance computing equipment, they have also been implemented in various embedded devices such as Arduino and FPGA [26, 27]. An attractive proposal to be able to implement

neural algorithms in embedded systems of low processing capacity is the *C-Mantec* algorithm, proposed in [28] and later implemented in an Arduino UNO board to analyse its characteristics [29]. However, in this study we will focus on the classic algorithms of the neural schemes: Perceptron, ADALINE, and the Hopfield network.

Table 1. Comparison of the technical characteristics of the Arduino UNO, Tiva-C and BeagleBone development boards.

Characteristic	Arduino UNO	Tiva- C	BeagleBone
Development place	Interaction Design Institute Ivrea, Italy.	Texas Instruments	Texas Instruments
Flash emory	32KB Flash memory.	256KB Flash Memory	2GB Flash Memory
Memory	Additional SD or Micro SD shield	32KB SRAM, 2KB EEPROM	512MB RAM, External microSD
Processing speed	16 MHZ	80Mhz	1GHz
Controller type	Atmega 328p	ARM Cortex-M4F, 32-Bit	ARM Cortex-A8 32-Bit
Internet connection	Additional shield, Wi-Fi	Adittional module, Wi-Fi	Ethernet port
General Purpose Input/Output (GPIO)	14 digital I / O pins (6 PWM outputs).	40	92
Software	Arduino IDE, external	Energia	Linux, Debian
Operating voltage	5 V	5 V (3.3 V, GPIO)	5 V
Average cost	\$ 22 USD	\$ 24 USD	\$ 50 USD

2.2.1 Perceptron

The perceptron, was proposed by *Frank Rosenblatt* in 1958 [30], this scheme can be visualized as the basic neuron within an ANN. It can only be used as a linear discriminator, that is, only linearly separable problems can be solved by a simple neuron. However, the interconnection of different neurons, manages to solve problems with complex behaviour such as can be seen in [31-34]. Figure 2 shows the basic scheme of a perceptron, while figure 3 shows a multilayer network, with several interconnected neurons.

2.2.2 ADALINE

Like the perceptron, the ADALINE is a simple neuron but with a linear response with respect to its inputs, it was designed to work with memistors [35].

This neuron was mainly differentiated from the perceptron by its learning rule, since the error is not adjusted after crossing the activation function, but the error is adjusted with respect to the sum of its inputs. ADALINE has been used in many applications, including [36-39]. The ADALINE scheme can be seen in figure 4.

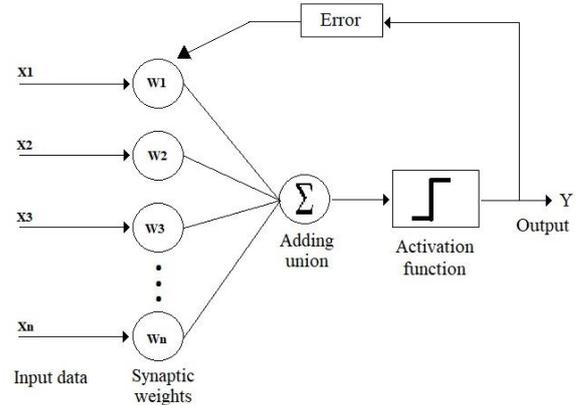


Fig. 2. Basic scheme of the perceptron.

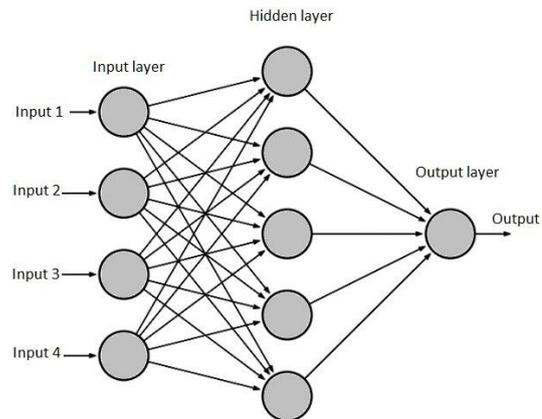


Fig. 3. Multi-layer perceptron.

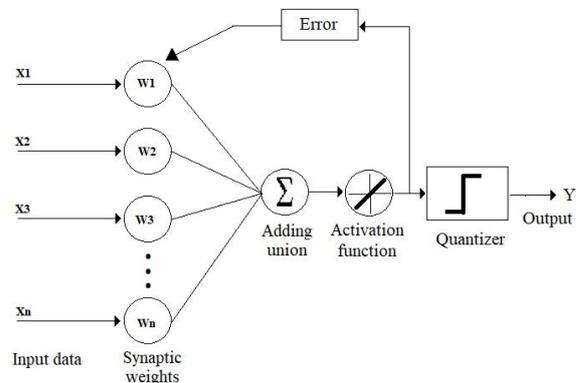


Fig. 4. Basic scheme of ADALINE.

2.2.3 Hopfield neural network

The *Hopfield neuronal network* was proposed in 1982 [40]. This network seeks to learn a set of patterns, and thus identify new input cases and associate them with one of the previously learned patterns. The neurons of this network, generally has only two states, activated and not activated. The Hopfield network is a single-layer network, but with a recurrent connection, which means

that each neuron is connected to all the other neurons, except itself. Some of the applications of this network can be found in [41-43]. The basic scheme of the Hopfield network can be seen in figure 5.

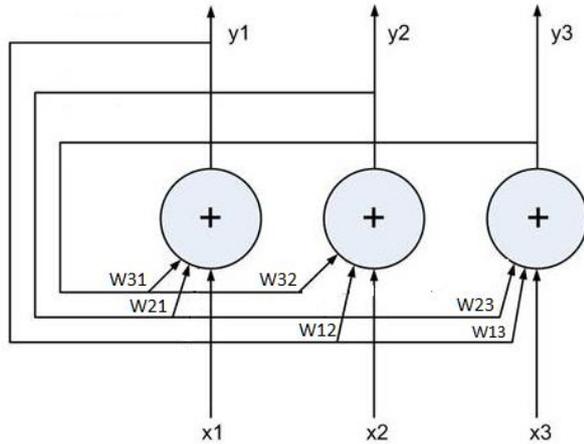


Fig. 5. Basic outline of the Hopfield network.

3 Development

This section shows the experiments carried out to obtain the time measurements when each of the aforementioned algorithms is executed in each of the development boards described above.

In the case of Arduino UNO and Tiva-C, there is a development environment from the PC in which the informatic program is made, and such program is loaded to each of the boards, so Arduino UNO and Tiva-C execute only the program related to the implemented algorithms.

On the other hand, the BeagleBone board is preloaded with the Debian OS, so it has a graphical interface. To access it, you can do it from its mini HDMI (High-Definition Multimedia Interface), or through a remote connection using a VNC server (Virtual Network Computing). In this case, the card has been used by means of the command line, using an SSH (Secure Shell) connection, for which the Putty client and the IP address 192.168.7.2 have been used.

For the implementation of the neural algorithms, each one of the mentioned schemes was executed in each one of the development boards, and the processing times of each executions were obtained.

3.1 Perceptron implementation

For the first experiment, a single execution of the Perceptron algorithm was performed on each of the development boards, although the algorithm stipulates that it starts with random W_i synaptic weights, for this first experiment, the initial synaptic weights were defined equal for each execution. This was done with the intention of having reference points. For the training, the logical AND function was performed, which can be seen in table 2.

Table 2. Training data of the simple perceptron, where we have two inputs X_1 and X_2 and the target of expected values Y .

X_1	X_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Once we had a single execution, the second experiment was performed, in which an average time of 100 Perceptron performances were obtained with the same initial values and with the W_i synaptic weights defined by the user.

As a third experiment, the same training scheme was presented, but with random W_i , obtaining the average time of 100 executions.

In Table 3, the execution times of each of the embedded systems can be observed, when the perceptron has been implemented once with defined W_i , the average time of 100 executions with W_i defined, and the average time of 100 executions with random W_i .

Table 3. Comparison of execution times when the perceptron is implemented in the different development boards. Given in millisecond (ms).

	Arduino Uno	Tiva-C LaunchPad	BeagleBone
One execution, defined W_i .	3 ms.	.047 ms.	.05809 ms.
Average time of 100 executions, defined W_i .	3 ms.	.047 ms.	.05807 ms.
Average time of 100 executions, random W_i .	3.1 ms.	.049 ms.	.05823 ms.

3.2 ADALINE implementation

For the implementation of the ADALINE, the calculation of a binary to decimal encoder was made, in table 4 the training scheme can be seen, where there are 3 inputs X_i and one expected output Y . The synaptic weights W_i have been defined by the user.

For the first experiment, a single execution of the algorithm was performed to measure the base time. In the second experiment, the average time of 100 executions with the same problem and with the same synaptic weights defined in the previous case was considered. And in the third experiment the average time of 100 executions performed with random initial values W_i was considered. In Table 5, we can observe the response time of this neural scheme within the three embedded systems.

Table 4. Input data and output target for the implementation of the ADALINE algorithm in the different development cards.

X_1	X_2	X_3	Y
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Table 5. Comparison of the execution times when the ADALINE scheme is implemented in the different development boards. Given in milliseconds (ms).

	Arduino Uno	Tiva-C LaunchPad	BeagleBone
One execution, defined W_i .	47 ms.	1.156 ms.	.677ms.
Average time of 100 executions, defined W_i .	46 ms.	1.134 ms.	.6763 ms.
Average time of 100 executions, random W_i .	49 ms.	1.176 ms.	.6932 ms.

3.3 Hopfield implementation

For the implementation of this algorithm, the neural network was trained with the figures shown in table 6. In the first two columns are the examples with which the network was trained, and in the third column is the element that was subsequently tried to rebuild it through the Hopfield network. It is a simple example, and the algorithm correctly reconstructed the test element.

Table 6. Samples with which the Hopfield network was trained, and the last column contains the element that was reconstructed.

Sample 1	Sample 2	Proof

For this case, only two experiments were carried out. In the first, the execution time of an execution is obtained, and in the second experiment the average time of 100 executions is obtained. The execution times can be seen in table 7.

Table 7. Execution times of the Hopfield algorithm implemented in each of the development boards. Given in milliseconds (ms).

	Arduino Uno	Tiva-C LaunchPad	BeagleBone
One execution.	.412 ms.	.035 ms.	.0516 ms
Average time of 100 executions.	.392 ms.	.034 ms.	.0517 ms

4 Analysis of results

Although Arduino UNO is the most commonly used board in different technological and development projects, it can be seen in tables 3, 5, and 7, Arduino has a much longer execution time with respect to the Tiva-C and BeagleBone boards. If we compare the Arduino UNO board with the Tiva-C, it can be seen in table 1, that the approximate prices of both boards are close to each other, however, the characteristics of processing speed and storage are very far from each other, comparing the prices, the speed of the board, and the results presented in tables 3, 5 and 7, it can be concluded that for the development of this type of neural network algorithms, Tiva-C board can be a better option than Arduino. One of the characteristics that makes the use of Arduino popular is the ease of programming, however, when using the Energia programming interface, the programming of Tiva-C and Arduino basically is done in the same way, as seen in figure 6.

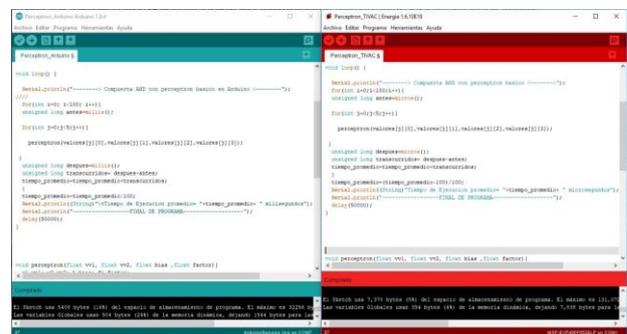


Fig. 6. Comparison between the Arduino IDE and Energia development interfaces. In both cases, the same perceptron execution code is running.

On the other hand, it can be seen that the execution times of the algorithms implemented between the BeagleBone and Tiva-C boards are not so far apart. In Table 1, it can be seen that the processing speed of the BeagleBone far exceeds the processing speed of the Tiva-C board, however, when the algorithms have been implemented, there is no such large difference between their execution times, we must mention that the BeagleBone is a board that runs an OS within itself, in this case Debian, so it performs the processing related to the OS, as well as the processing of the algorithms executed at the same time.

If processes are not very complex, considering speed and ease, Tiva-C can be a very good option. If more complex processes have to be executed, BeagleBone can be used.

5 Conclusions

As it has been presented throughout this paper, neural algorithms for processing data are widely used in different types of projects, many of them carried out in systems that have high processing and storage power. However, the new schemes of communication and

interaction between electronic devices through the internet, require that the obtaining, processing and transmission of information be carried out in devices that can be portable, such as the different development boards that exist in the market.

One of the main disadvantages of these development boards is their limited processing and storage capacity. However, such boards can be efficiently used to execute algorithms with few data.

In this paper we have presented the implementation of the Perceptron, ADALINE and Hopfield neural network algorithm in the Arduino UNO, Tiva-C and BeagleBone development boards, to measure their execution times and determine which of the boards presented in this study will be used as the best option for the generation of communication and processing systems when you have a limited amount of data.

If the comparison of the Arduino board with Tiva-C is made, it can be seen that the Tiva-C LaunchPad development board presents a better response at the execution time of the implemented algorithms, better technical characteristics, and has the same programming ease when the environment Energia is used.

In the carried out experiments, it was found that the Tiva-C development board is a good option for the design of projects that may be within the IoT scheme, since it has a much shorter execution time compared to the Arduino UNO, currently the most used.

The BeagleBone board has a longer processing speed with respect to the Arduino UNO card, however, BeagleBone processes the calculations related to the implemented neural algorithms, and the Debian OS with graphic interface at the same time, ie, its computational resources are not only destined to the processing of the input data, but also they are destined to the OS.

Acknowledgments: This research was funded by UNAM projects: PAPIIT IN 105219, PAPIIME PE111519, PIAPIME 4.31.05.19.

References

1. Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Mao, M.; Senior, A.; Tucker, P.; Yang, K.; Le, Q. V Large scale distributed deep networks. NIPS (2012).
2. Sajjad, M.; Nasir, M.; Muhammad, K.; Khan, S.; Jan, Z.; Sangaiah, A.K.; Elhoseny, M.; Baik, S.W. Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities. *Futur. Gener. Comput. Syst.* (2017).
3. De Coninck, E.; Verbelen, T.; Vankeirsbilck, B.; Bohez, S.; Simoens, P.; Demeester, P.; Dhoedt, B. Distributed neural networks for internet of things: The big-little approach. *In Proceedings of the Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*; (2016).
4. Leroux, S.; Bohez, S.; De Coninck, E.; Verbelen, T.; Vankeirsbilck, B.; Simoens, P.; Dhoedt, B. The cascading neural network: building the Internet of Smart Things. *Knowl. Inf. Syst.* (2017).
5. Pavithra, D.; Balakrishnan, R. IoT based monitoring and control system for home automation. *In Proceedings of the Global Conference on Communication Technologies, GCCT 2015* (2015).
6. Kumar, N.S.; Vuayalakshmi, B.; Prarthana, R.J.; Shankar, A. IOT based smart garbage alert system using Arduino UNO. *In Proceedings of the IEEE Region 10 Annual International Conference, Proceedings/TENCON*; (2017).
7. Chianese, A.; Piccialli, F. Designing a smart museum: When cultural heritage joins IoT. *In Proceedings of the Proceedings - 2014 8th International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST 2014*; (2014).
8. He, N.; Qian, Y.; Huang, H.W. Experience of teaching embedded systems design with BeagleBone Black board. *In Proceedings of the IEEE International Conference on Electro Information Technology*; (2016).
9. Zhong, X.; Liang, Y. Raspberry Pi: An Effective Vehicle in Teaching the Internet of Things in Computer Science and Engineering. *Electronics* (2016).
10. Bader M. O. Al-thobaiti, Iman I. M. Abosolaiman, Mahdi H. M. Alzahrani, S.H.A.A.; Mohamed S. Soliman* Design and Implementation of a Reliable Wireless Real-Time Home Automation System Based on Arduino Uno Single-Board Microcontroller. *Int. J. Control. Autom. Syst.* (2014).
11. Fransiska, R.W.; Septia, E.M.P.; Vessabhu, W.K.; Frans, W.; Abednego, W.; Hendro Electrical power measurement using Arduino Uno microcontroller and LabVIEW. *In Proceedings of the Proc. of 2013 3rd Int. Conf. on Instrumentation, Communications, Information Technol., and Biomedical Engineering: Science and Technol. for Improvement of Health, Safety, and Environ., ICICI-BME* ; (2013).
12. Juang, H.S.; Lurr, K.Y. Design and control of a two-wheel self-balancing robot using the arduino microcontroller board. *In Proceedings of the IEEE International Conference on Control and Automation, ICCA*; (2013).
13. Nayyar, A.; Puri, V. A review of Beaglebone smart board's-a Linux/android powered low cost development platform based on ARM technology. *In Proceedings of the Proceedings - 9th International Conference on Future Generation Communication and Networking, FGCN*; (2016).
14. Siradjuddin, I.; Tundung, S.P.; Indah, A.S.; Adhisuwigno, S. A real-time Model Based Visual Servoing application for a differential drive mobile robot using Beaglebone Black embedded system.; (2016).

15. Chianese, A.; Piccialli, F.; Riccio, G. Designing a smart multisensor framework based on beaglebone black board. *In Proceedings of the Lecture Notes in Electrical Engineering*; (2015).
16. Desai, N.S.; Alex, J.S.R. IoT based air pollution monitoring and predictor system on Beagle bone black. *In Proceedings of the 2017 International Conference On Nextgen Electronic Technologies: Silicon to Software, ICNETS2 2017*; (2017).
17. Shariff, S.U.; Swamy, J.C.N.; Seshachalam, D. Beaglebone black based e-system and advertisement revenue hike scheme for Bangalore city public transportation system. *In Proceedings of the Proceedings of the 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology, iCATccT*; (2017).
18. Benadda, B.; Elgorma, M.; Beldjilali, B. Embedded BeagleBone based Wi-Fi intrusions detector and vulnerabilities checker. *In Proceedings of the 2017 Seminar on Detection Systems Architectures and Technologies, DAT 2017*; (2017).
19. Morales, M. An Introduction to the Tiva™ C Series Platform of Microcontrollers 2013, Texas Instruments, (2013).
20. Torres Galindo, A.K.; Gómez Rivera, A.F.; Jiménez López, A.F. Development of a multispectral system for precision agriculture applications using embedded devices. *Sist. Telemática Orig. Res.* (2015).
21. Ahmed, S.; Shakev, N.; Milusheva, L.; Topalov, A. Neural net tracking control of a mobile platform in robotized wireless sensor networks. *In Proceedings of the Proceedings of the 2015 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics, ECMSM 2015*; (2015).
22. Mason, C.H.S.; William, Y.; Albert, F.Y.C.; Darmandran, S. Design and implementation of sub-GHz wireless light switch with integrated Wi-Fi. *In Proceedings of the IEEE International Conference on Communication, Network and Satellite IEEE COMNETSAT*, (2014).
23. Abeyruwan, S., Sikder, F., Visser, U., & Sarkar, D. Activity Monitoring and Prediction for Humans and NAO Humanoid Robots Using Wearable Sensors. *In Proceedings of the FLAIRS Conference*; 2015; pp. 342–347. (2015)
24. Buckley, J.J.; Hayashi, Y. Fuzzy neural networks: A survey. *Fuzzy Sets Syst.* (1994).
25. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* (2017).
26. Ortega-Zamorano, F.; Jerez, J.M.; Munoz, D.U.; Luque-Baena, R.M.; Franco, L. Efficient Implementation of the Backpropagation Algorithm in FPGAs and Microcontrollers. *IEEE Trans. Neural Networks Learn. Syst.* (2016).
27. Ortega-Zamorano, F.; Molina-Cabello, M.A.; López-Rubio, E.; Palomo, E.J. Smart motion detection sensor based on video processing using self-organizing maps. *Expert Syst. Appl.* (2016).
28. Subirats, J.L.; Franco, L.; Jerez, J.M. C-Mantec: A novel constructive neural network algorithm incorporating competition between neurons. *Neural Networks* (2012).
29. Ortega-Zamorano, F.; Jerez, J.M.; Subirats, J.L.; Molina, I.; Franco, L. Smart sensor/actuator node reprogramming in changing environments using a neural network model. *Eng. Appl. Artif. Intell.* (2014).
30. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* (1958).
31. Sadegh, N. A Perceptron Network for Functional Identification and Control of Nonlinear Systems. *IEEE Trans. Neural Networks* (1993).
32. Pandey, P.C.; Barai, S. V. Multilayer perceptron in damage detection of bridge structures. *Comput. Struct.* (1995).
33. Orhan, U.; Hekim, M.; Ozer, M. EEG signals classification using the K-means clustering and a multilayer perceptron neural network model. *Expert Syst. Appl.* (2011).
34. Yan, H.; Jiang, Y.; Zheng, J.; Peng, C.; Li, Q. A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Syst. Appl.* (2006).
35. Widrow, B. Adaptive "adaline" Neuron Using Chemical "memistors.". (1960).
36. Chen, C.I.; Chang, G.W. A two-stage ADALINE for harmonics and interharmonics measurement. *In Proceedings of the Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications, ICIEA 2010*; (2010).
37. Abdel-Galil, T.K.; El-Saadany, E.F.; Salama, M.M.A. Power quality event detection using Adaline. *Electr. Power Syst. Res.* (2003).
38. Joorabian, M.; Mortazavi, S.S.; Khayyami, A.A. Harmonic estimation in a power system using a novel hybrid Least Squares-Adaline algorithm. *Electr. Power Syst. Res.* (2009).
39. Zhang, G.; Wang, G.; Xu, D.; Zhao, N. ADALINE-network-based PLL for position sensorless interior permanent magnet synchronous motor drives. *IEEE Trans. Power Electron.* (2016).
40. Hopfield, J.J. Neural networks and physical systems with emergent collective abilities. *Proc. Natl. Acad. Sci. U. S. A.* (1982).
41. Zhu, Y.; Yan, H. Computerized tumor boundary detection using a Hopfield neural network. *IEEE Trans Med Imaging* (1997).
42. Tatem, A.J.; Lewis, H.G.; Atkinson, P.M.; Nixon, M.S. Super-resolution target identification from remotely sensed images using a Hopfield neural

- network. IEEE Trans. Geosci. Remote Sens. (2001).
43. Tatem, A.J.; Lewis, H.G.; Atkinson, P.M.; Nixon, M.S. Super-resolution land cover pattern prediction using a Hopfield neural network. Remote Sens. Environ. (2002).