|  |  |  |
| --- | --- | --- |
| **First semester report** | **July 2018** | **Note:** |
|  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Student to complete this section | | | | | | |
| **Matusowsky** | **M.** | **Mr** | **11093804** | | | |
| **Design of a virtual sensor using machine**  **learning imputation techniques in a**  **wireless sensor network.** | | | **Study leader:**  **Mr D. Ramotsoela** | | | |
| Degree programme enrolled for: | **Project number:** | | |  | Revision number: | 0 |
| **Student declaration**:  I understand what plagiarism is and that I have to complete my project on my own. I am fully aware of the University’s policy in this regard. I have not used another student’s past or present written work to submit as my own. I declare that this report is my own original work. Where other people’s work has been used (either from a printed source, internet or any other source), this has been properly acknowledged and referenced in accordance with departmental requirements. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_  Student signature Date | | | | | |

|  |  |
| --- | --- |
| Declaration by language editor | |
| I have been allowed adequate time to read this document carefully and to make corrections where necessary(date received indicated below).  To the best of my knowledge, correct formatting, spelling and grammar are used throughout the document. | |
|  |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  L.K. Malan (language editor) | \_\_\_\_\_\_\_\_\_\_\_  Date |

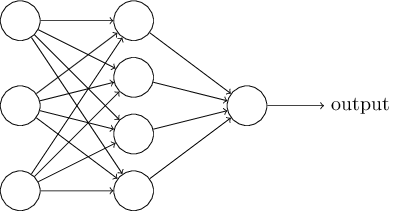
|  |
| --- |
| **Table of contents** |

|  |  |
| --- | --- |
| 1. Literature study | 2 |
| 2. Approach | 3 |
| 3. Project plan | 5 |
| 4. Progress | 7 |
| 5. References | 8 |

|  |
| --- |
| **1. Literature study** |

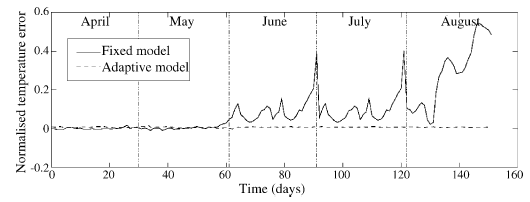
To be able to design the proposed system, it is necessary to understand the principles of artificial neural networks and the various methods that are available to train them in order to potentially deploy one as a virtual sensor. This paragraph will detail some of the neural networks that have been implemented in the literature with regards to environmental temperature sensing.

Artificial neural networks are a network of simple elements called artificial neurons which receive one or more inputs and then are passed through an activation function to change the internal state to produce an output based on the input and activation function. The multi-layered perceptron (MLP) neural network consists of a minimum three layers of neurons namely the input layer, the hidden layer (of which there may be more than one in the case of deep neural networks) and the output layer which are fully connected and each connection having its own weight. There are various methods to train the MLP weights with two methods being popular today namely the backpropagation algorithm [1] and genetic algorithms [2] . A typical representation of neural network is shown in **Figure 1** below showing that every layer is fully connected to the next layer.



**Figure 1: A basic neural network** [3]**.**

There have been multiple implementations of neural networks with regards to weather prediction and room temperature sensing. Devi et al [4] implemented a feed-forward neural network that would be used to predict weather patterns given multiple environmental inputs. They use backpropagation as the training method due to the ability of the algorithm to capture the complex relationship between the many factors that might influence the environmental temperature which may include but not be limited to: Atmospheric pressure, wind speed, wind direction, humidity, dew point temperature and elevation above sea level. The model was able to infer a relationship between the given inputs and outputs of historical data suggesting that a neural network can be used to predict some weather patterns however the system was not deployed in a real-time environment as the research was focused on the viability of a neural network prediction model. Hayati and Mohebi [5] applied a 3-layer neural network to design a short-term temperature forecasting system for Kermanshah city, Iran. Ten years of weather forecasting data from 1996-2006 was used to train the MLP, specifically 6-hour average soil temperature at various depths up to 100 cm, wet and dry temperatures [6], humidity, pressure, hours of sunshine and radiation. Training was done using the scaled conjugate gradient method, a numerical optimization technique, described by [Møller](https://www.sciencedirect.com/science/article/pii/S0893608005800565" \l "!) [7]. The appropriate number of hidden neurons used was determined through an iterative trial and error process where hidden neurons were added and the MLP re-initiated to a random state after a certain amount of epochs until a few MLP models emerged as likely candidates for predictive weather forecasting. The downside to this is that a lot of time is needed to run through the iterative process; however, the training method used converges much faster than the backpropagation algorithm making it a still viable method of determining the optimal MLP [8]. The optimal hidden neurons was found to be six, after 2000 epochs. The mean absolute error minimum and maximum of 0.0079 and 1.2916 show that the MLP is very accurate in terms of predicting future temperature based on recent historical data, never deviating more than 1.5 *°* C from the actual measures temperatures. Ruano et al [9] implemented a MLP system in a school for smart energy systems to save on electrical costs. The costs involved expensive on-site meteorological tools and equipment and indoor sensors placed throughout the school. The type of sensors involved air temperature, humidity, atmospheric pressure sensors, the state of the doors and windows (open or closed) and air-conditioner power consumption. It is important to note that all data was acquired without human noise or noise from equipment i.e. a perfect environment. The MLP is trained using the Levenberg-Marquardt method [10, 11] due to it regressing faster than backpropagation and a second attempt is made using genetic algorithms due to the inability of the algorithm to fully explore the model space. Using the genetic algorithm, 1000 MLPs were initiated and 100 generations were run where 100 candidate models per generation would repopulate the population for the following generation. The MLP performs quite well when used in the period of the year that the training data was gathered (April-May) but decreased in accuracy as the weather changed from autumn to summer where no previous data was collected to train the MLP. An illustration of this error is shown in **Figure 2**.



**Figure 2: Neural network error increase as the year proceeds from the trained period** [9]**.**

This project aims to fill the gap in the literature where a MLP is deployed on multiple sensor nodes and on a central server to account for more than one room or area of a room and deployed in real-time. Neural networks have not been deployed on sensor nodes in previous literature but have run off a central server.

|  |
| --- |
| **2. Approach** |

The sensor nodes can be implemented using various different types of microcontrollers. A PIC32 was decided on as the microcontroller of choice as the processing power is deemed more than sufficient to deal with the computations required by the neural network which will be implemented both on the sensor nodes and on the server.

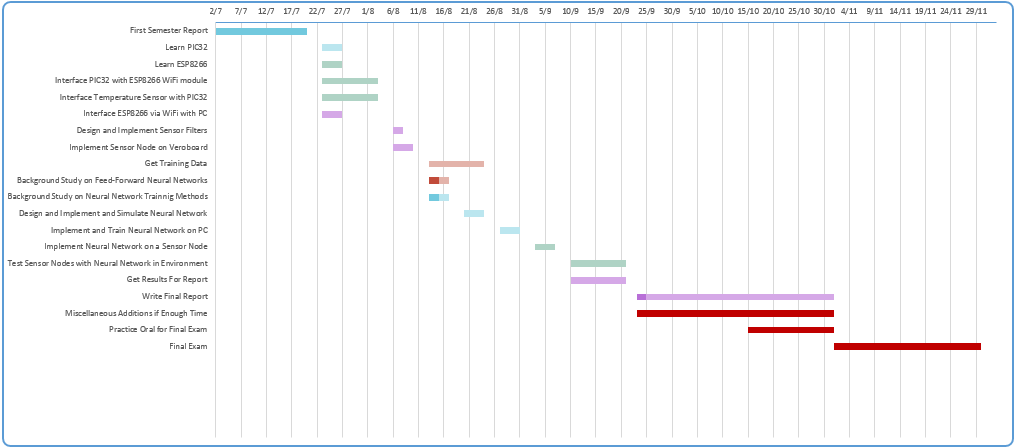
The sensors can be filtered for noise using either physical analog filters or through the use of digital filters. While the processing power of the PIC32 would allow the implementation of digital filters along with the neural networks, analog filters will be investigated for viability due to concerns regarding the programming memory of the PIC32.

The wireless communication protocol between the sensor nodes and the server can be implemented using Bluetooth or WiFi communication standards. The advantages of both these communication protocols is that the hardware modules are both relatively cheap in terms of cost per module and are easy to configure and use in a 1-to-1 configuration. The Bluetooth modules however require being paired before being used and would require compliance with the Bluetooth 4.1 standard [12] in order to be used in a star network (many-to-1) topology whereas the WiFi modules would only require an established and available TCP/IP network which can be as simple as a cellphone hotspot or a university WiFi network in order to communicate in a star network topology with the server. Furthermore it was decided that there will be a checksum check implemented on both the server and sensor modules to ensure that the integrity of the received data is maintained when information is transmitted or received over the network.

The neural network models that will be implemented and deployed will be determined on an iterative basis where many different models will be trained and evaluated for performance based on the number of hidden neurons in a layer and the number of hidden layers as well as the amount of training epochs required to sufficiently train the neural network. Training can be done using a genetic algorithm or backpropagation. The advantage of using a genetic algorithm is that it will be capable of optimising the model using a much wider search space and is less prone to regressing to a local minima like the backpropagation algorithm. The disadvantage is that training time for a genetic algorithm is significantly higher than backpropagation. However, due to the necessity to evaluate the performance of multiple neural networks of different parameters, the genetic algorithm is the most likely candidate that will be used for training purposes.

|  |
| --- |
| **3. Project plan** |

**Table 1** below shows the project plan with a resolution of 5 days. The plan details the estimated amount of time required per task and includes a set amount of time for possible extra additions not specified in the original project proposal.



**Table 1: Project Plan**

|  |
| --- |
| **4. Progress** |

|  |
| --- |
| **5. References** |

# References

|  |  |
| --- | --- |
| [1] | C. Macleod, An Introduction to Practical Neural Networks and Genetic Algorithms for Engineers and Scientists, Aberdeen: Robert Gordon University, 2001. |
| [2] | W. Banzhaf, P. Nordin, R. Keller and F. Francone, Genetic Programming – An Introduction, San Francisco: Morgan Kaufmann, 1998. |
| [3] | M. A. Neilsen, Neural Networks and Deep Learning, Determination Press, 2015. |
| [4] | C. J. Devi, B. S. P. Reddy, K. V. Kumar, B. M. Reddy and N. R. Nayak, "ANN Approach for Weather Prediction," *International Journal of Engineering Trends and Technology,* vol. 3, no. 1, pp. 19-23, 2012. |
| [5] | M. Hayati and Z. Mohebi, "Application of Artificial Neural Networks for Temperature Forecasting," *International Journal of Electrical and Computer Engineering,* vol. 1, no. 4, pp. 662-666, 2007. |
| [6] | Engineering ToolBox, "Dry Bulb, Wet Bulb and Dew Point Temperatures," 2004. [Online]. Available: https://www.engineeringtoolbox.com/dry-wet-bulb-dew-point-air-d\_682.html. [Accessed 17 July 2018]. |
| [7] | M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks,* vol. 6, no. 4, pp. 525-533, 1993. |
| [8] | Y. Linde, A. Buzo and R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications,* no. 28, pp. 84-95, 1980. |
| [9] | A. E. Ruano, E. M. Crispim, E. Z. E. Conceic¸a˜o and M. M. J. R. Lu´cio, "Prediction of building’s temperature using neural networks models," *energy and buildings,* vol. 38, no. 6, pp. 682-694, 2006. |
| [10] | K. Levenberg, "A method for the solution of certain problems in least," *Quarterly Applied Mathematics,* vol. 2, pp. 164-168, 1944. |
| [11] | D. Marquardt, "An algorithm for least-squares estimation of nonlinear," *SIAM Journal of Applied Mathematics,* vol. 11, pp. 431-441, 1963. |
| [12] | "Bluetooth," Wikimedia Foundation, Inc, [Online]. Available: https://en.wikipedia.org/wiki/Bluetooth#Bluetooth\_4.1. [Accessed 18 July 2018]. |