SFU SURJ SIMON FRASER UNIVERSITY SCIENCE UNDERGRADUATE RESEARCH JOURNAL Vol. 1 (2015-16)

Neural networks and their application in the assessment of the healthiness of the human sitting posture

Ahmad Al Attar[1]*, Chingiz Jakubaliyev[1], Haitham Kaddoura[1], Mohammad Azarbar[1]

[1]Simon Fraser University, *Department of Mechatronic Systems Engineering*

Abstract

Sitting posture can have a significant effect on an individual's health. Machine learning can be used to assess the "healthiness" of a given sitting posture. A target equation was conceived and applied with input data into a neural network learning algorithm. The resulting trained neural network was successful in assessing the healthiness of the sitting postures of four hypothetical cases. The neural network design brings us one step closer to an artificially intelligent chair that can alert users if their sitting postures are unhealthy.

Keywords — Neural Networks, Machine Learning, Ergonomics, Sitting Posture, Unhealthiness Value

1. Introduction

Sitting for too long or in an improper way can lead to various health issues. Sitting posture is an important factor that can significantly affect the health of an individual [1]. Ergonomics is the study of human efficiency in the workplace. Several ergonomically-designed chairs exist that promise healthier seating during work. However, not all ergonomic chairs match everyone. Selecting the right chair can be difficult. It is important to find the chair that best suits one's body and activity for a healthy work style [2]. Machine learning neural networks can be used to design and program an artificially intelligent system that learns from a set of previously collected data. The trained neural network is then able to assess the healthiness of a given sitting posture. In this research, sitting posture data was generated and target data was calculated from a conceived mathematical equation that describes the unhealthiness level of a given posture. Both data sets were used to train the neural network. The neural network was then tested on new data and assessment results were obtained.

2. Theory

2.1. Machine Learning

Machine learning is a branch of artificial intelligence in which computers are programmed to modify and adapt their algorithms so that they become more accurate.

*Corresponding Author: info.machmse@gmail.com

There are three main types of learning: *supervised*, *unsupervised*, and *reinforcement learning*. See Figure 1. Supervised learning is very similar to giving a student a problem set along with the solutions to study and prepare for a quiz. The premise of this approach is that the student generalizes the ideas contained in the problem set. Unsupervised learning is similar to giving the student the problem set without solutions and expect the student learns to generalize the ideas to prepare for the quiz. Lastly, reinforcement learning is somewhere in-between supervised and unsupervised learning. In reinforcement learning, the student gets told when the answer is wrong, but does not get told how to correct the mistakes [3].

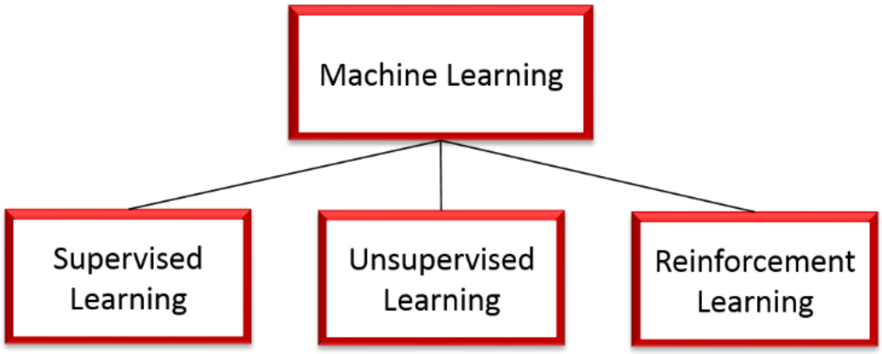

Figure 1: *The three types of machine learning.*

2.2. Neural Networks

Neural networks are a branch of supervised machine learning and cognitive science that are inspired by the neural networks in the human nervous system and brain. There are a lot of benefits to learning about these models and their utility in intelligent systems. Since it is a form of supervised learning, the "problem set" along with the "solutions" must be provided to help the neural network generalize and prepare answers to new questions. The elementary unit that processes the computations within the nervous

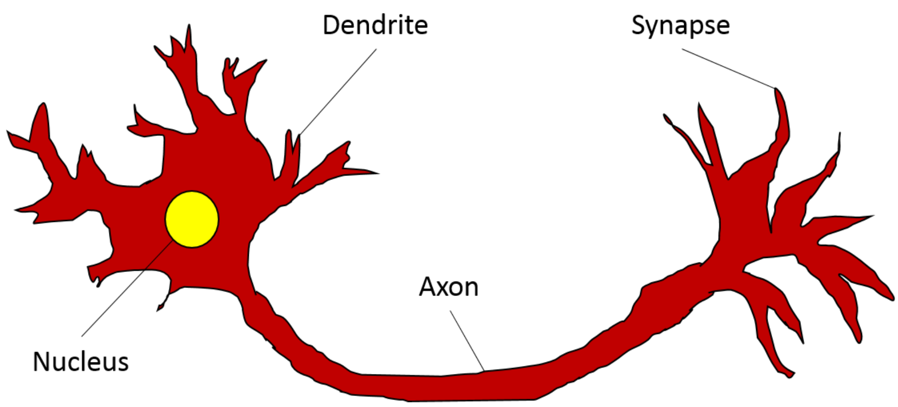

Figure 2: *Biological neuron.*

system of a human is the neuron. Similarly, the elementary unit that processes the computations within an artificial neural network is the perceptron. Before diving into the details of these models, biological neurons, from which we build our model of the perceptron, will be briefly introduced. In the human brain, there are about 10^{11} neurons that can communicate with each other. Each neuron has hundreds or even thousands of connections to other neurons in the human brain [4]. See Figure 2 for a figure of the generic neuron along with some key terminologies.

For our interest, only two features provide the basis for artificial neural networks: dendrites and synapses. The dendrites serve as the input terminals of a neuron. Synapses are junctions at the output of a neuron which also serve as input to other neurons. It is important to note that in neural network theory, it is believed that synapses vary in strength. Connections between some neurons are enhanced while between others are inhibited [5]. In an over simplified example, let's say a certain set of neurons contain the image of your aunt, and if your aunt gives you a banana every time you visit her, then the connections between the neurons that contain your aunt's image and the neurons that contain the image of a banana grows stronger. Now, the next time you see your aunt, you will instantly think of bananas.

A neuron receives inputs and then it either fires a signal or not. This is neatly modelled by the McCulloch-Pitts perceptron. See Figure 3.

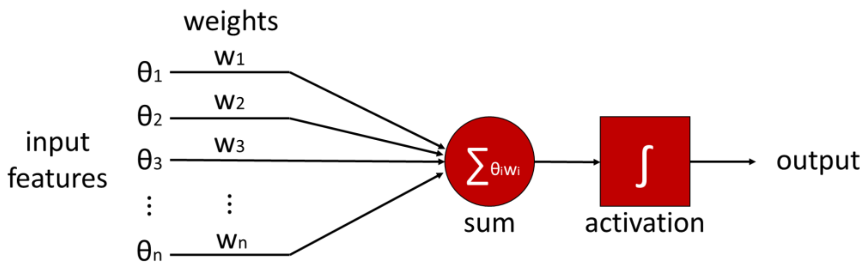

Figure 3: *McCulloch-Pitts perceptron.*

The inputs are multiplied by weights to account for the fact that their connection strengths vary. Whether a perceptron fires or not is determined by the weighted sum of all inputs. If the sum exceeds a certain threshold function, it fires, otherwise it does not emit any signal [6]. Instead of this 1 or 0 output caused by the choice of activation function, one can pick different activation functions. The most commonly used linear activation function generates a continuum of values. As shown in Figure 3, a single perceptron has many inputs and generates a single output. Just like a system of neurons build up a brain, a system of perceptrons build up a neural network. See Figure 4 for a simple neural network (the weights have been dropped out of the figure for clarity).

We see that all the inputs are fed into all the perceptrons and the number of outputs matches the number of perceptrons (which matches the number of input features). The column that has the inputs is usually called the *input layer* and the column that has the outputs is usually called the *output layer*. It is often useful to have a *hidden layer* of perceptrons in between the input and output layers that aid in performing more complex machine learning. See Figure 5.

So far, we have discussed the basic working principles of a neuron, introduced the computational analogy of a neuron, the perceptron, and looked at how neural networks

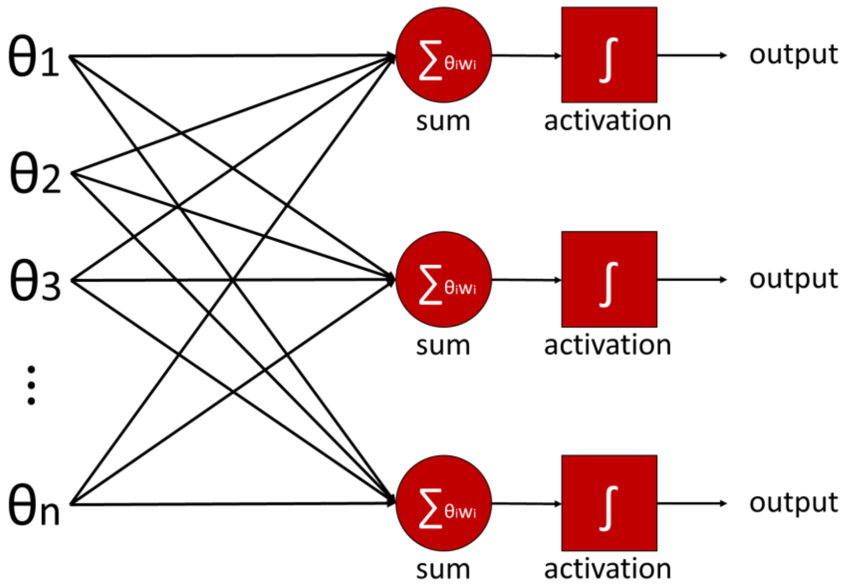

Figure 4: *Simple neural network.*

can be built from a system of perceptrons. Only one fundamental ingredient is left: the learning part of the model. How does a neural network learn? Or even starting simpler, how does a perceptron learn? Let us think of the four key features that make up a perceptron. The input comes from a set of previously collected data, the output either also comes from previously collected data (in case of training) or it is to be figured out (in case of testing), and the activation function is selected before running the algorithm. In other words, the input is set, the output is set, and the activation function is set. Thus, learning happens at the weights as shown in Figure 6.

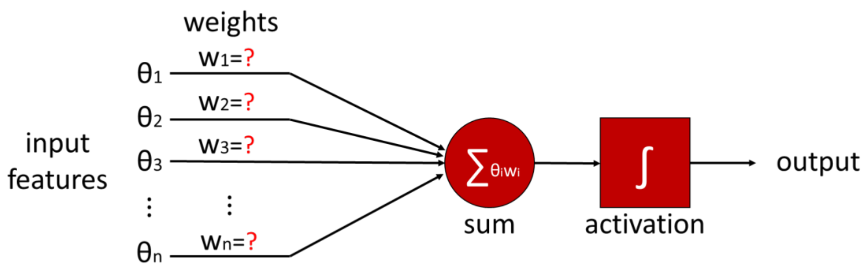

Figure 6: *Weights to be determined during training stage.*

In brief, the algorithm starts by setting the weights to small random numbers. During the training stage, it then finds the output using those weights and compares it to the target (solution) and then comes up with better weights to match the target more accurately. This training technique, called backpropagation, results in a set of weights that allows the neural network to produce the correct outputs to the inputs used in training. It is now ready to be tested on new inputs. The neural network provides an effective method to create intelligent systems that can learn.

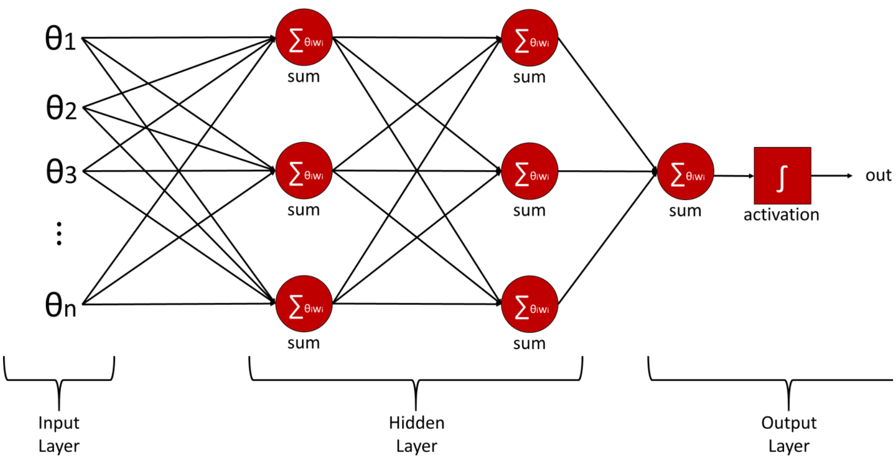

Figure 5: *Multilayer neural network.*

2.3. Input Data

In our research, angles in a given sitting posture are randomly generated. These angles will then serve as the input set to our artificially intelligent algorithm. These featured angles are listed in Table 1 which leads to assumption 1, which is presented in the appendix.

Table 1: *List of input features of interest.*

Input Features of Interest			
1. Knee Angle	3. Trunk Inclination Angle	5. Hip Angle	7. Upper Arm Angle
2. Ankle Angle	4. Elbow Angle	6. Leg Crossing	8. Lower Arm Angle

Each of these features have a healthy range which is measured by the Canadian Centre for Occupational Health and Safety. Figure 7 shows the configurations of the healthy ranges of the sitting posture features or feature angles listed Table 1. Table 2

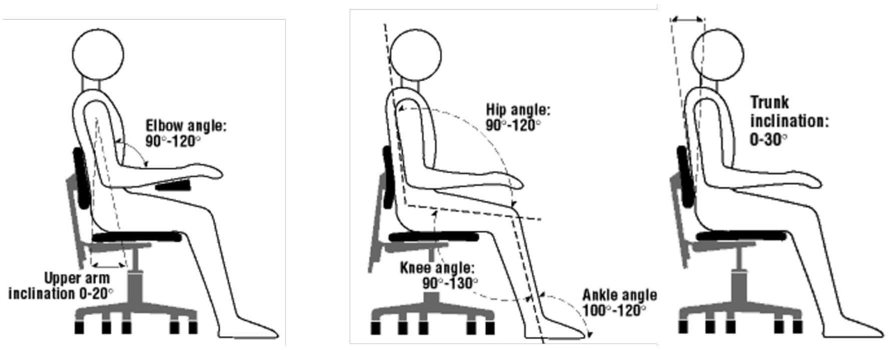

Figure 7: *Healthy ranges in sitting posture [7].*

summarizes the healthy ranges of the sitting posture features that are of interest.

Table 2: *Healthy ranges for sitting posture features of interest.*

Sitting Posture Feature	Healthy Angle Ranges (degrees)
Knee Angle	90–130
Ankle Angle	100–120
Trunk Inclination	0–30
Elbow Angle	90–120
Hip Angle	90–120
Forearm Angle	0–20
Upper Arm Inclination	0–30

Some people tend to cross their legs while sitting. Crossing legs has three types which are:

Ankle-to-Ankle Ankle-to-Knee Knee-to-Knee

Ankle-to-Ankle means resting an ankle on another ankle. Ankle-to-Knee is when a subject rests an ankle on a knee. Knee-to-Knee means resting a knee on the other knee [8]. For each feature, an "Unhealthiness Value" is to be assigned. The unhealthiness value or level is a number that suggests how unhealthy a given posture angle is, leading to the second assumption. The unhealthiness values of the leg crossing types were assumed to be those shown in Table 3.

Table 3: *Unhealthiness values of leg crossing.*

Leg Crossing	Unhealthiness Value
No Crossing	0
Ankle-to-Ankle	10
Ankle-to-Knee	20
Knee-to-Knee	30

2.4. Target of Learning Stage

Since neural networks are a type of supervised learning, the targets (solutions) must be provided along with the inputs for the algorithm to learn to generalized and prepare itself to give answers to new inputs. The supervised nature of neural networks divides the assessment process in two stages: training and testing. In the training stage, the algorithm is trained by being shown the inputs and targets (the problems and solutions). In the testing stage, the algorithm is shown new inputs (problems only) from which it needs to intelligently provide targets (answers) to. The features mentioned in Table 1 serve as the inputs. The question we seek to answer is: in the training stage of the neural network, what are the targets that are fed into the algorithm along with the inputs?

We already know that an input is simply a collection (in fact, a vector) of feature angles (such as upper arm angle, lower arm angle, etc.). However, what is the target?

In other words, we have the problem set, what are the solutions so the algorithm can learn to generalize?

To get a good sense of the unhealthiness of a given sitting posture, we make use of the fact that within a given range of angles a certain feature (such as upper arm angle) is considered healthy. As we deviate from this healthy range the unhealthiness level increases. Figure 8 portrays this idea [leading to assumption 4].

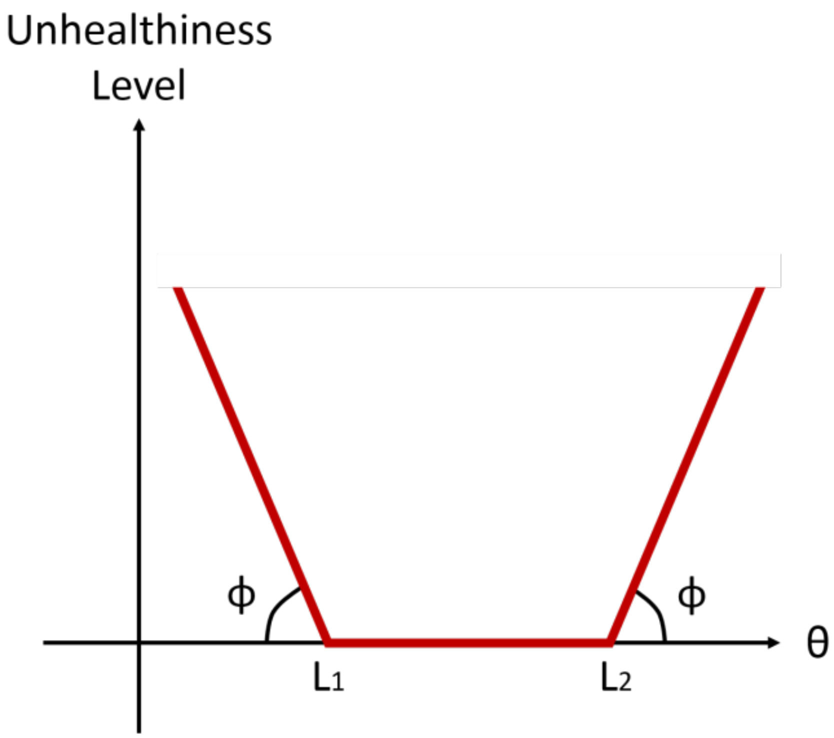

Figure 8: *Unhealthiness level of a particular sitting posture feature angle.*

We notice that the unhealthiness level of an angle that is within the healthy range is zero. If there are N features, then there are N of such plots, each of which will give us an unhealthiness value for a given feature angle. A vector is constructed from a set of angles measured from an individual. This vector serves as the input set. Each angle in the vector is then converted to an unhealthiness value. The obtained values are mapped to a range from 0 to 10. The sum of these numbers can also be mapped to a range from 0 to 10, yielding the Overall Unhealthiness Value, R. Thus, a given sitting posture in which all the features are within the healthy range will have $R = 0$. The mathematical equation that is used to generate the targets for a given input (both of which are used for the learning stage of the network) is presented in Equation 1 [which leads to assumption 5].

$$\sum_{i=1}^{n} y_i \rightarrow \{0 \leq y \leq 10\} = R = \text{Unhealthiness Value} \tag{1}$$

where

$$\begin{aligned}
&-\tan\phi(\theta_1 - L_{1,1}) + \tan\phi(\theta_1 - L_{1,1})u(\theta_1 - L_{1,1}) \\
&\qquad + \tan\phi(\theta_1 - L_{1,2})u(\theta_1 - L_{1,2}) \rightarrow \{0 \leq y \leq 10\} = y_1, \\
&-\tan\phi(\theta_2 - L_{2,1}) + \tan\phi(\theta_2 - L_{2,1})u(\theta_2 - L_{2,1}) \\
&\qquad + \tan\phi(\theta_2 - L_{2,2})u(\theta_2 - L_{2,2}) \rightarrow \{0 \leq y \leq 10\} = y_2, \\
&\qquad\vdots \\
&-\tan\phi(\theta_n - L_{n,1}) + \tan\phi(\theta_n - L_{n,1})u(\theta_n - L_{n,1}) \\
&\qquad + \tan\phi(\theta_n - L_{n,2})u(\theta_n - L_{n,2}) \rightarrow \{0 \leq y \leq 10\} = y_n
\end{aligned}$$

where θ_n is the nth input feather, ϕ is the intensity, $L_{n,1}$ and $L_{n,2}$ are the limits of the healthy region, (from Figure 8, and $u(\theta_n)$ is the unit step function [see assumption 6].

Since we are looking at only 7 features (see Table 1 or Table 2), n will be 7 (for leg crossing unhealthiness values, refer to Table 3). This target gives a good sense of how unhealthy a given sitting posture is. Now, we are able to train the neural network (given that we now have inputs and their corresponding targets that are the two essential components for the training stage of any supervised machine learning algorithm). Figure 9 presents a compact illustration of the training and testing stages of the algorithm.

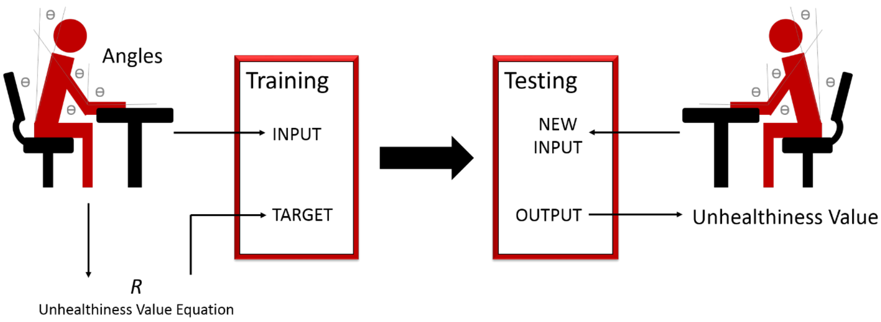

Figure 9: *Training and testing stages of machine learning algorithm.*

Note that the input data was not collected from actual individuals, but was randomly generated instead in order to answer the question, "*Can we use neural networks along with the target function conceived above to have the algorithm learn to assess new test cases?*" within the timeframe available.

3. Method

The neural network was programmed in Matlab. The code took an input matrix and a sample matrix. The input matrix was made up of column vectors of sitting posture angles. Each vector had 8 instances. In other words, each column represents a single individual and each row is a feature – such as knee angle, lower arm angle, etc. The angles were randomly generated also in Matlab. The code uses this input matrix to generate a target vector using the target function conceived (see Figure 8). Both the input matrix and target vector were used to train the neural network. The neural network consisted of 10 hidden layers. This neural network is illustrated in Figure 10.

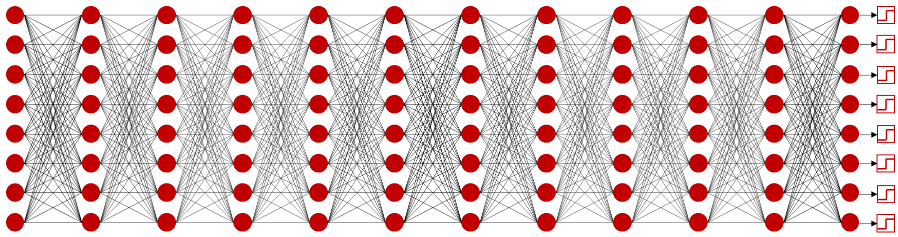

Figure 10: *Neural network with 10 hidden layers.*

After training the neural network, test cases (test postures) were assessed to see whether or not the neural network has successfully generalized to the data and is able to assess the unhealthiness level of given sitting postures.

4. Results and Discussion

Input data were randomly generated that represent the sitting posture angles of 135 individuals in Matlab. The result is shown in Figure 11. The x-axis represents the 8 feature angles (in the order listed in Table 1).

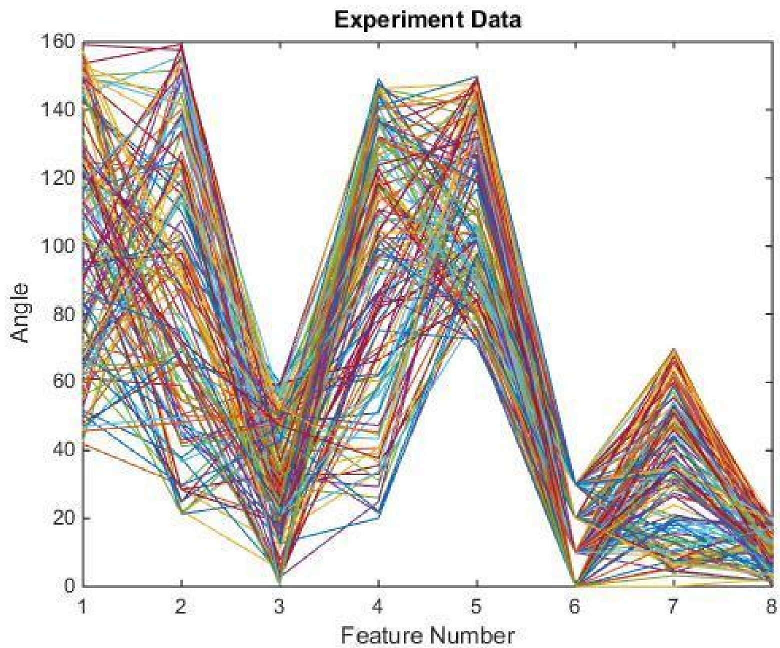

Figure 11: *Randomly generated input data that represent the 8 features of 135 individuals.*

The input data was used to create the target vector (using the target function) that contained unhealthiness values of each individual in the input matrix. Both, the input matrix and the target vector, were inputted into the neural network constructed using the Neural Network Toolbox in Matlab. The plots below in Figure 12 show that the slope of the curves are almost 1, which indicates that the algorithm has successfully learned from the input.

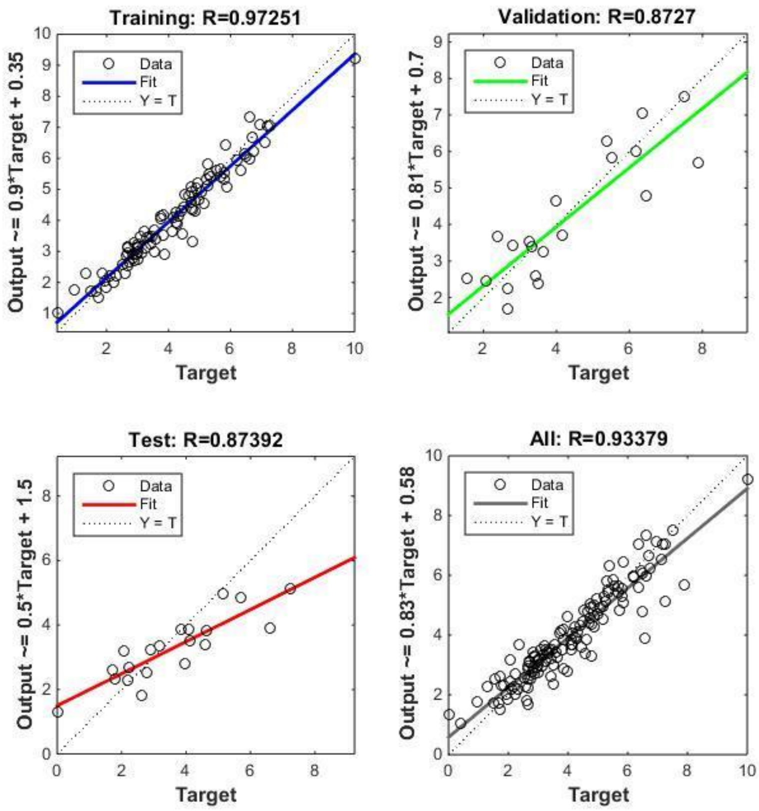

Figure 12: *Neural network training results.*

Next, we tested the algorithm using four test cases. The test set is tabulated in Table 4. The four columns represent four hypothetical subjects. The first and second column contain feature angles all within the healthy ranges. However, the third column represent an unhealthy (bad) case and the fourth column represents the worst of the four.

Inputting those values into the neural network and prompting for the Unhealthiness Values, we obtained 0, 0, 3, 10 for the four hypothetical subjects, respectively. These results matched the theory we developed. It indicated that the first two subjects are sitting in a healthy posture while the third is sitting in an unhealthy posture. The fourth subject is the unhealthiest with respect to the other three. The neural network successfully gave results that help in the assessment of the healthiness (or unhealthiness) of sitting postures.

Table 4: *Neural network testing subjects.*

Individual	1st	2nd	3rd	4th
Knee Angle	100	120	140	20
Ankle Angle	120	115	130	20
Trunk Inclination Angle	20	10	40	60
Elbow Angle	100	110	130	150
Hip Angle	100	110	135	150
Crossed Leg	0	0	10	30
Upper Arm Angle	10	15	30	60
Lower Arm Angle	10	5	25	50

5. Conclusion

A neural network consisting of 10 hidden layers was used to assess the healthiness of an individual's sitting posture. The Unhealthiness Value was used as a measure of how unhealthy an individual's sitting posture is. A set of input data was randomly generated in Matlab. The targets were generated using Equation 1. Both the input data and the targets were input to the neural network. The algorithm was programmed in Matlab. Results showed that the neural network successfully learned from the input data and corresponding targets. The algorithm was then tested on four hypothetical cases of sitting postures. The neural network successfully assessed the unhealthiness level of the four cases. Thus, neural networks can be used to assess the healthiness of human sitting posture. The algorithm can be used to learn from real complex data and be integrated in smart chairs that might alert the user to sit in a healthy manner. It is important to note that the algorithm has limitations due to the fact that the healthiness of sitting postures depend on various aspects that were not considered in this research. These include age of the individual, duration of sitting, and physical injuries [7, 9, 10].

References

[1] James A Levine. Sick of sitting. *Diabetologia*, 58(8):1751–1758, 2015. URL http://link.springer.com/article/10.1007/s00125-015-3624-6.

[2] Canadian Centre for Occupational Health and Safety. Ergonomic chair, 2014. URL https://www.ccohs.ca/oshanswers/ergonomics/office/chair.html.

[3] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC press, 2015.

[4] Suzana Herculano-Houzel. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience*, 3:31, 2009. doi:10.3389/neuro.09.031.2009.

[5] James A Anderson. *An introduction to neural networks*. MIT press, 1995.

[6] Samantha Hayman. The mcculloch-pitts model. In *Neural Networks, 1999.*

IJCNN'99. International Joint Conference on, volume 6, pages 4438–4439. IEEE, 1999. doi:10.1109/IJCNN.1999.830886.

[7] Canadian Centre for Occupational Health and Safety. Good body position, 2010. URL `http://www.ccohs.ca/oshanswers/ergonomics/sitting/sitting_position.html`.

[8] Soonjae Ahn, Seunghyeon Kim, Sunyoung Kang, Hyeseon Jeon, and Youngho Kim. Asymmetrical change in the pelvis and the spine during cross-legged sitting postures. *Journal of Mechanical Science and Technology*, 27(11):3427–3432, 2013. doi:10.1007/s12206-013-0865-5.

[9] MedlinePlus. Aging changes in the senses, 2014. URL `https://www.nlm.nih.gov/medlineplus/ency/article/004013.htm`.

[10] Solbi Kim, Yoonhee Chang, and Gyoosuk Kim. Correlations between biomechanical characteristics, physical characteristics, and the ability to maintain dynamic sitting balance on an unstable surface in the disabled with spinal cord injury. *Journal of the Ergonomics Society of Korea*, 33(1):15–25, 2014. doi:10.5143/JESK.2014.33.1.15.

6. Appendix

6.1. Assumptions

1. Features that are vital to the healthiness of a sitting posture are assumed to be only the ones listed in Table 1.

2. Features are assumed to have a corresponding Unhealthiness Value or Level depending on their angle.

3. The unhealthiness values for leg crossing postures are assumed to be those listed in Table 3.

4. The unhealthiness value of any feature is assumed to follow Figure 8.

5. The target is assumed to follow Equation 1.

6. All features are assumed to have equal intensity ϕ.