

# **The Instantiation of Brains Design Document**

## **Final project**

Matthew A. Letter  
&  
Lin sun

University of New Mexico

Neural Networks 547

mletter1@unm.edu  
sun@unm.edu

# Abstract: left blank intentionally

## Table of Contents

<b>Abstract .....</b>	<b>3</b>
<b>1. Introduction .....</b>	<b>4</b>
<b>2. Description of Anticipated Designs.....</b>	<b>4</b>
2.1 Architecture 0: No movement, measure lifetime.....	4
2.2 Architecture 1: Movement, measure lifetime as a function of speed.....	4
2.3 Architecture 2: Movement and eat-all-on-contact, measure lifetimes as a function of speed .....	4
2.4 Architecture 3: Movement, eat all on contact, classification with Delta Rule Neuron for one eyelet, plot RMS training error versus eat event .....	5
2.5 Architecture 4-10: Movement based on light and sound inputs with additions of the Back Propagation algorithm. ....	5
<b>3. Learning Approach .....</b>	<b>5</b>
3.1 Initial approach .....	5
3.2 Learning.....	6
3.3 Combining the learning.....	6
<b>4. Evaluation Approach:.....</b>	<b>6</b>
4.1 Testing the robot/brain in the environment.....	6
4.2 Collected Data.....	7
4.3 Criteria for success of the project.....	7
<b>5. Analysis/Presentation Approach: .....</b>	<b>7</b>
5.1 Analysis.....	7
5.2 Presentation .....	8
<b>6. Preliminary Results .....</b>	<b>8</b>
6.1 Architecture 0: No movement, measure lifetime.....	8
6.2 Architecture 1: Movement, measure lifetime as a function of speed.....	9
6.3 Architecture 2: Movement and eat-all-on-contact, measure lifetimes as a function of speed .....	10
6.4 Architecture 3: Movement, eat all on contact, classification with Delta Rule Neuron for one eyelet, plot RMS training error versus eat event .....	10
<b>7. Known Issues .....</b>	<b>12</b>
7.1 Architecture 3 possible design issue .....	12
7.2 Future Milestones.....	12
<b>8. References .....</b>	<b>12</b>

## **Abstract**

Add abstract here, single space, 12pt intentionally blank

## **1. Introduction**

This paper rigorously establishes that neural network design models can be used to manipulate learning on simulated biological system. All the research was done using professor Thomas Caudell's animal robot environmental model. (Caudell) The starting point of the research involved 3 architectures, used to establish the parameters of life with respect to the simulated organism and its environment. These provide a basis for analyzing the implemented neural network algorithms, the goal of which is to create a "neuron" based brain for the organism to live as long as possible in its environment, and determine what neuronal design structures degrade or enhance this attribute.

## **2. Description of Anticipated Designs**

### 2.1 Architecture 0: No movement, measure lifetime

In this design, the robot animal was set to stay in one place until it dies. In other words, forward speed of the robot is set as 0 in Controller.c file.

### 2.2 Architecture 1: Movement, measure lifetime as a function of speed

In this design, the robot animal was set to move in a direction until it dies. The direction was incremented by 5° after every epoch.

### 2.3 Architecture 2: Movement and eat-all-on-contact, measure lifetimes as a function of speed

In this design, the robot animal was set to move in a direction, eating everything in its path, until it dies. The direction was incremented by 5°, after every epoch. The goal of which is to establish whether there is good food and bad food on the board.

## 2.4 Architecture 3: Movement, eat all on contact, classification with Delta Rule Neuron for one eyelet, plot RMS training error versus eat event

In this design, the robot animal was set to move in a direction, eating everything in its path, until it dies. The direction was incremented by 5°, after every epoch. The goal of which is to classify good food and bad food on the board based on the LMS neuron architecture. It is expected that this architecture will be used later for movement based on food classification from light input. The algorithm behind this architecture model is to eat all food during movement, and adjust the neuron weights based on the light input and energy change after eating. It will produce a linear function of 0 or 1 based on the light input (RGB) values.

## 2.5 Architecture 4-10: Movement based on light and sound inputs with additions of the Back Propagation algorithm.

In these future designs, the robot animal will move based on light and sound inputs and the energy change after eating. The goal of which is to either move the animal or have it stay in place based on its own energy and what it sees, listens and smells in its environment. All subsequent architectures will take a little step in this general direction (all steps will be based on the results of the previous steps). In a predictive fashion, architecture 4 will expand what the animal sees moving from one forward input direction of light to all 31 input light vectors and associate them with the Back Propagation algorithm, architecture 5 will take architecture 4's ability to see food in all directions of one and apply it to both eyes, architecture 6 will play off the previous 2 and try to get the robot to change directions based off what it sees. The next potential architectures, will try to incorporate what was gleaned from the eyes and apply it to smell as well. The overall encompassing project goal, of these architectures, will be to keep the animal alive as long as possible.

## **3. Learning Approach**

### 3.1 Initial approach

Our initial approach involves making small incremental model design steps that can be leveraged by the next model iteration, depending on energy change.

### 3.2 Learning

The goal of the first “real” neuron architecture (3) is simply to classify food as either good or bad (neutral is not included here); based on the net change in energy after eating. This data will be tied in with the RGB vector, for incoming light straight ahead of the robot, and used to update weights for classifying objects.

Moving down this line of logic, after the robot can correctly classify food, we can use the food classification and association with light vectors to determine a direction to move. For example, if the robot sees food coming in from one of its RGB vectors, the robot can re-orient in the direction of the good food based off the output of the neuron for food classification.

### 3.3 Combining the learning

More neurons can be added to the system based on the light vector food classifier making a good setup for a Back Propagation multilayer feedback network. In this network, we plan on implementing our LMS neuron into a Back Propagated network, after which it is planned that we will apply the same set of logic to the sounds and smell sensors. The goal of which is that at the end of these architecture builds is to combine them all to determine the direction in which the mouse moves (or does not move). Once this is set up we can add another neuron into the loop that determines if the robot should stay in one spot or move towards food based off whether it has lots of energy or needs to go feed.

## **4. Evaluation Approach:**

### 4.1 Testing the robot/brain in the environment

The overall test of the robot brain will be to see if we can push the lifespan of our robot greater than any of the naive architectures (0-2) using the neural algorithms described in subsections 3.2 and 3.3.

#### 4.2 Collected Data

Ideally we would like to do over 1000 runs for each set of new data collection. This will be done with every architecture model. Data will be collected and stored after every epoch to be used in plots, table, and graphs to emphasize any changes a new architecture presents. RMS error data will be collected for all neuron architectures. Furthermore any significant information “learned” by the neurons will be save for presentation, such as wavelengths for RGB vectors of poisonous and non-poisonous foods.

#### 4.3 Criteria for success of the project

Our success will be determined by being able to make an accurate decision, based on the data, that there were either positive or negative changes in the robots life span, for each architecture model. The main goal of this project is to lengthen the lifespan of the robot animal. Also, being able to rule out bad architectures during the experiments will be considered successful.

### **5. Analysis/Presentation Approach:**

#### 5.1 Analysis

Histograms, averages, and standard deviations, for each experimental condition in each architecture model, will be used to aid analysis. Also, depending on the neural implementation, things such as types and quantities of food being eaten and average speed will be track, or other interesting key data.

## 5.2 Presentation

In order to make analysis run smoothly, all experimental neural models will plot RMS error and life span data. Histograms, scatter-plots, world maps, lifespan graphs and many other data point will be plotted in matlab or excel in order to gain knowledge of what advantage and/or disadvantage each new model presents. Plots that exemplify good or bad points will be used to re-enforce or discredit any new neuronal iteration of the robot. All other “less” key plotted data will be attached to the appendix for reference. The results given in the report can either be bad or good, as they still reinforce the underlying goal presented earlier in section 1.

## **6. Preliminary Results**

### 6.1 Architecture 0: No movement, measure lifetime

Table 1: Over 200 runs, every run of a stationary robot had a lifespan of 5001 cycles

Number of Runs	Lifespan of Robot
200	5001 cycles



## 6.2 Architecture 1: Movement, measure lifetime as a function of speed

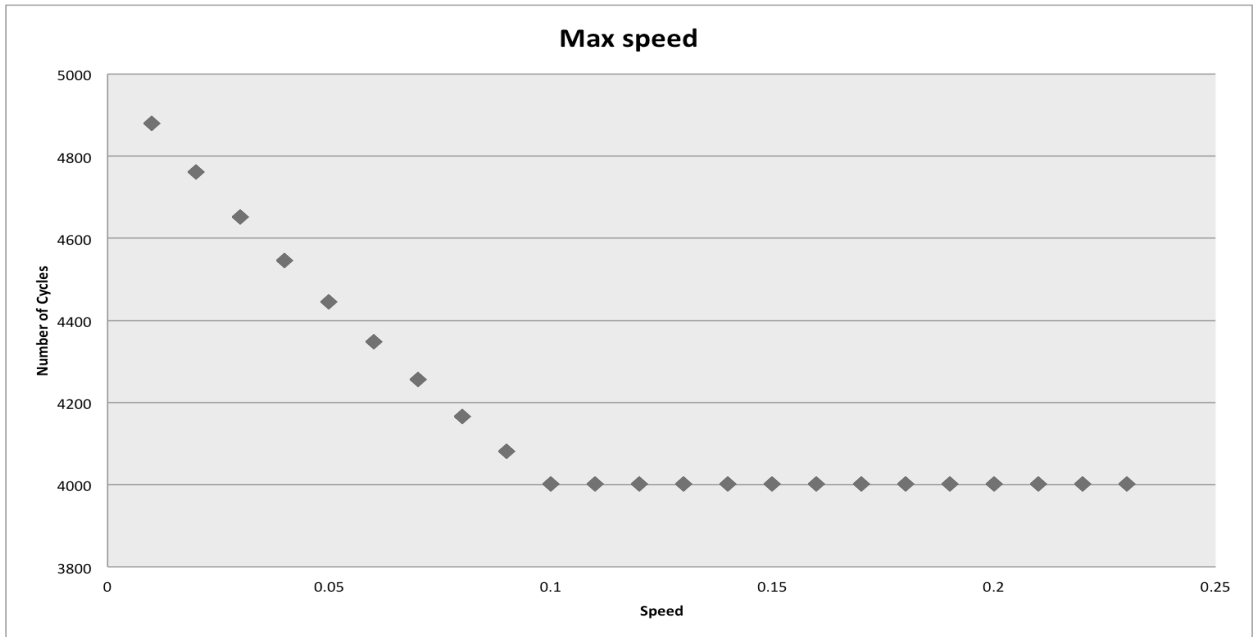


Figure 1: Shows the max speed of the robot to be 0.1

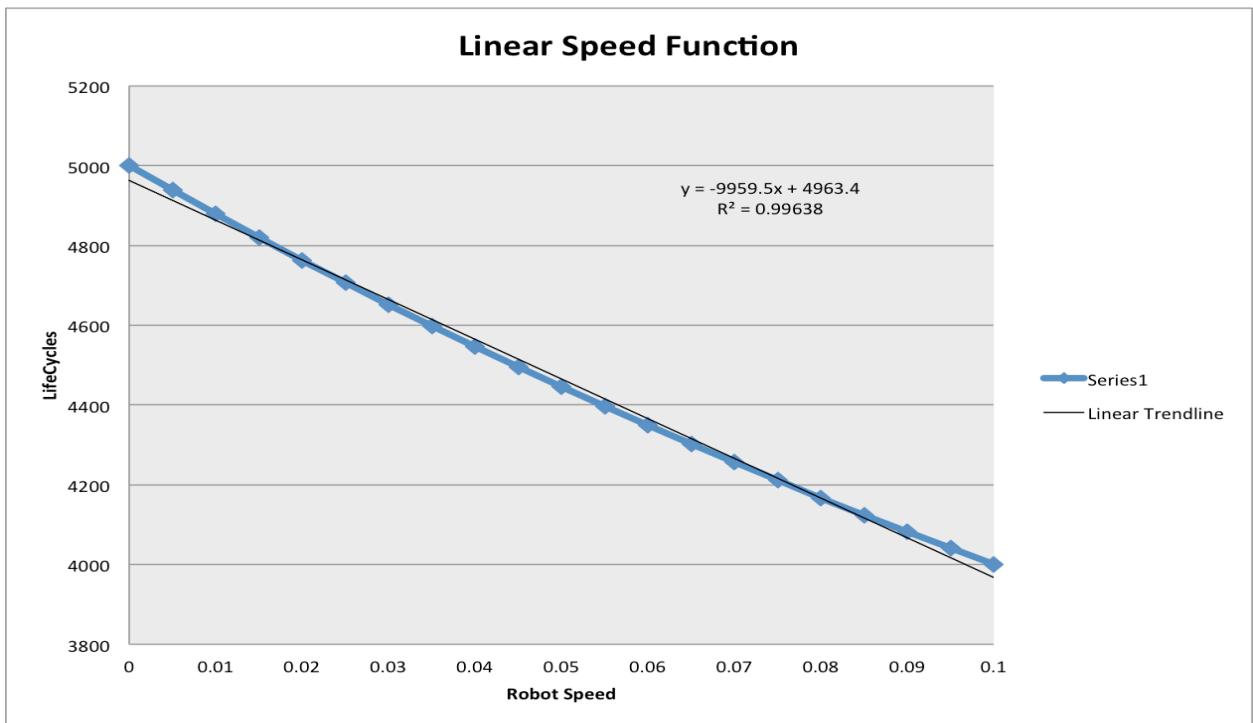


Figure 2: Linear function of the lifecycles with respect to the Robots speed. The standard deviation was 0.

### 6.3 Architecture 2: Movement and eat-all-on-contact, measure lifetimes as a function of speed

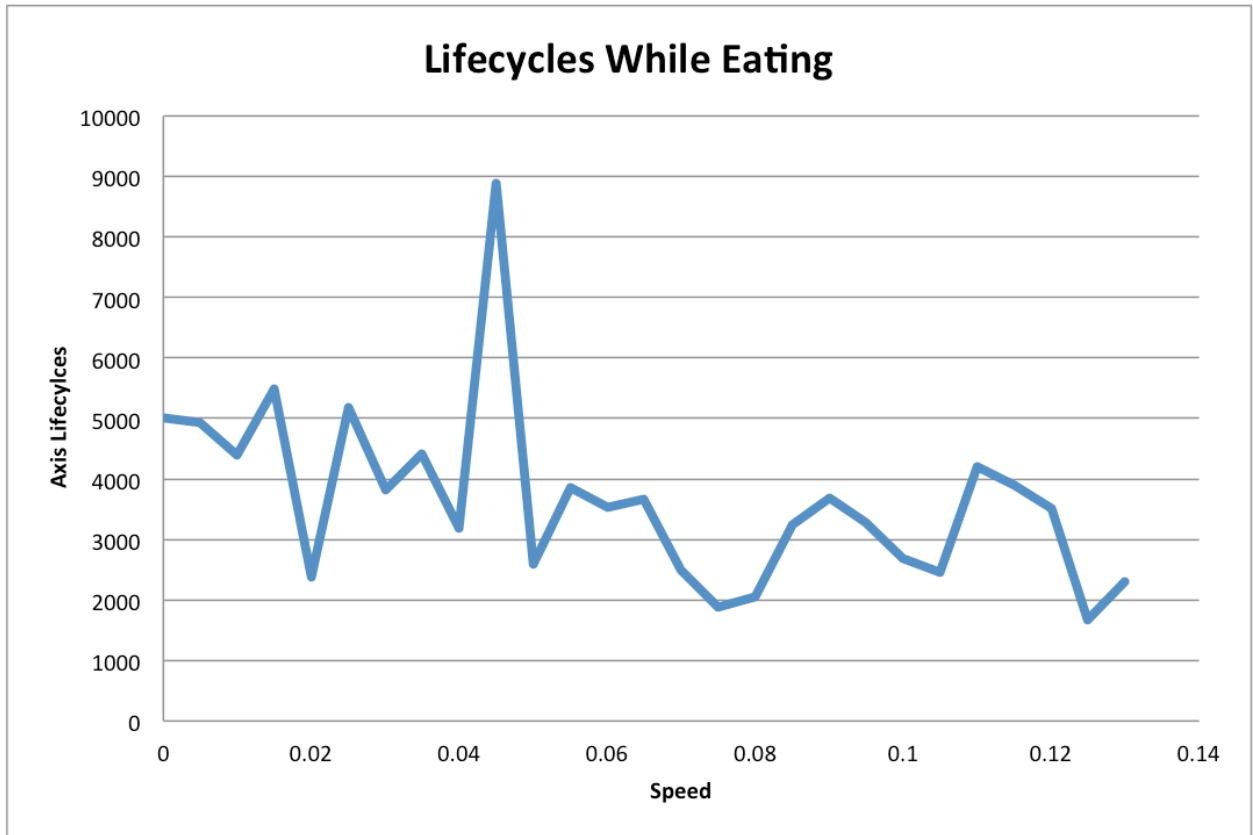


Figure 3: Eating while varying the movement speed. We see that things in the world can either hurt the robot or help the robot.

### 6.4 Architecture 3: Movement, eat all on contact, classification with Delta Rule Neuron for one eyelet, plot RMS training error versus eat event

In this design, only eating events which cause energy change will be used for neuron classification. Instead of setting output target as the real energy change, we will use 0 or 1 based on positive or negative energy change after eating. If eating event causes an increase in energy, the expected output should be 1. Otherwise, the expected output is set as 0. Figure 5 and Figure 6 show the RMS change during each run respectively. Table 2 shows the final training weights when neuron dies for 5 runs.

Table 2: Final weights

	W0	W1	W2	W3
1	-0.07393	-0.11732	1.00157	0.430708
2	0.0154733	-0.268242	0.879703	0.736229

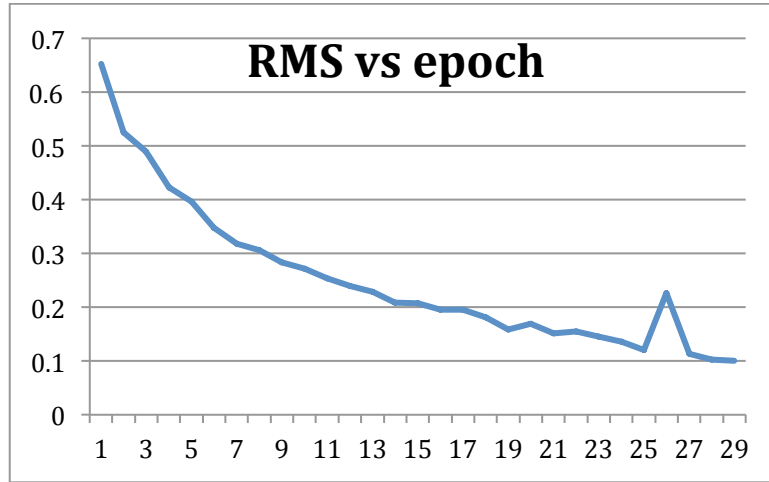


Fig 5

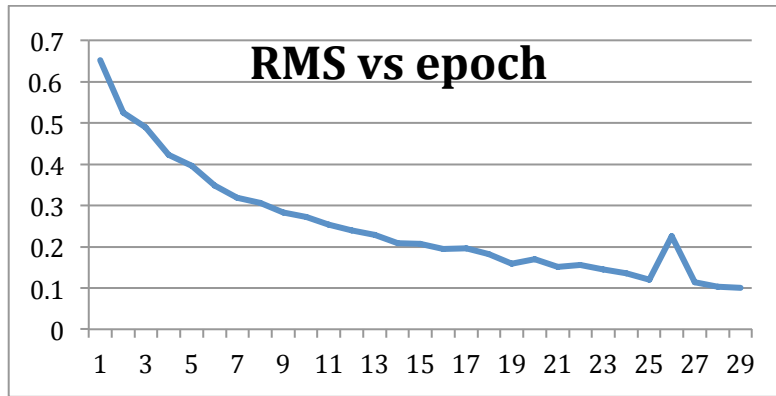


Fig 6

Descriptions of the first four architectures.- Preliminary performance results for each of these.

## **7. Known Issues**

### 7.1 Architecture 3 possible design issue

1. In Architecture 3, only one neuron is used to classify food and neutral food is not included in classification. The classification result could be different if neutral food is added into classification.

### 7.2 Future Milestones

Our team goal is to complete one new Architecture model a week, up to the last two weeks of class. In the last two weeks of class we plan to build our final paper.

## **8. References**

Caudell, Thomas. "Flatworld."