**Application Of Neural Networks To Robot Animals**

**Final Project CS547**

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

Abstract 3

1. Introduction 4

2. Approach 4

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, measure lifetime as a function of speed 4

2.4 Architecture 3: Neuronal Classification of food 5

2.5 Architecture 4: Neuronal Movement towards brightest object (prove that algorithm same as winner takes all network) 5

2.6 Architecture 5: Eat objects based off neuronal classification 5

3. Results 5

3.1 Architecture 0: No movement, measure lifetime 5

3.2 Architecture 1: Movement, measure lifetime as a function of speed 6

3.3 Architecture 2: Movement, measure lifetime as a function of speed 7

4. Discussion 8

5. Summary and Conclusion 8

6. Acknowledgements 8

7. References 8

8. Appendix A 8

# 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 robot organism to live as long as possible in its environment, and determine what neuronal design structures degrade or enhance this attribute.

# 2. Approach

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

For the first architecture, we will test the basic metabolic consumption for the robot. So we just set the robot’s speed as 0 and do nothing until it dies. Then we record down how many cycles the neuron could survive for.

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

This test will try to explore the effect of robot speed on robot’s lifespan. So that we could find a speed where the speed change will not affect the final result. Here we will let neuron start at the same position, but in different direction. To try test different direction, we will turn the neuron head to 5 degree counter-clockwise each time. When the neuron is moving, it will always head to the brightest object in the space, no matter if the food is poisonous or not. Each time after the neuron finish moving, it will eat nothing in the space, so that the only variable in this test is the neuron speed.

### 2.3 Architecture 2: Movement, measure lifetime as a function of speed

This test will try to explore the effect of neuron speed on robot’s lifetime. However, it is different from Architecture 2 in that neuron will eat food it means each time. What is more, we will not let neuron to classify object this time. Just let it eats whatever it contacts. Therefore, we could know more about speed effect based on eating. When the neuron is moving, it will always head to the brightest object in the space, no matter if the food is poisonous. To get the speed effect randomly, we will change the initial neuron head angle each time, but keep the initial position the same as origin point.

### 2.4 Architecture 3: Neuronal Classification of food

This test will try to classify food as either good or bad. An LMS neuron will be use to classify the food. A gain in health points will be classified as good and a loss in health point will be classified as bad. The light vector from the food will be used as inputs.



Figure 1: Network diagram of the LMS neuron for classifying food.

### 2.5 Architecture 4: Neuronal Movement towards brightest object

In order to increase the interaction with the Robots environment and to give the robot the ability to seek out objects the following design was produced.

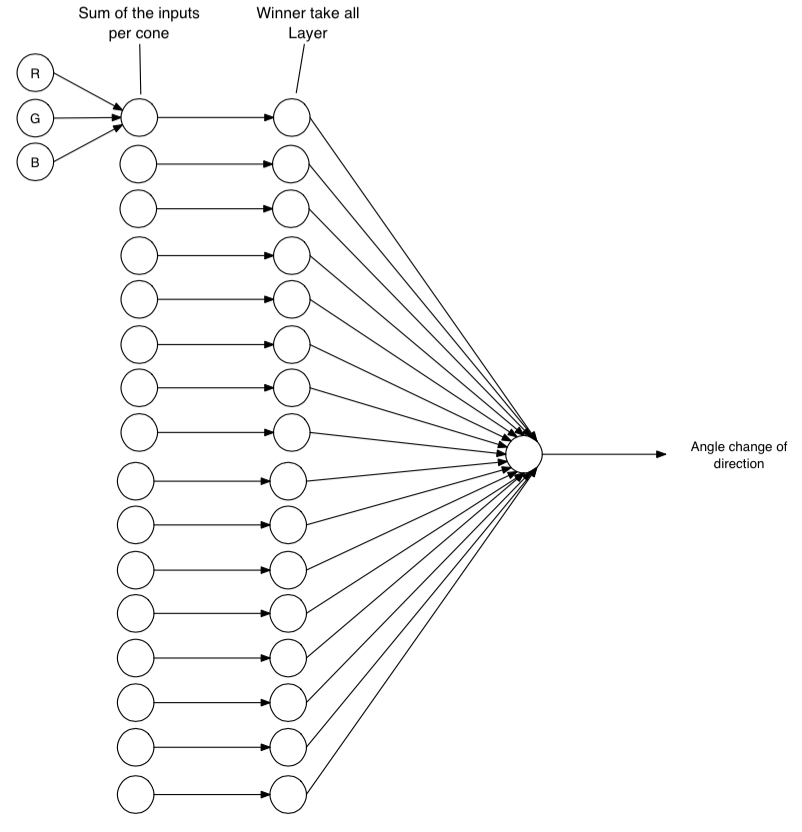


Figure 2: This is a three layer neural network where the RGB circle denotes the intensity signal bands being received by the first layer. The second layer is a winner-take all network which selects the brightest cone. The third layer is a neuron, which computes the angle change for movement

### 2.6 Architecture 5: Eat objects based off neuronal classification

# 3. Results

### 3.1 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 | Standard Deviation |
| 200 | 5001 cycles | 0 |

From Table 1, we could see that the robot life is 5001 cycles if it does not move and does not eat any food in the space. From settings in the system, we could see that the total initial charge of 1. The robot needs one cycle to jump out the loop. So totally there should 5000 cycles to consume energy in the system. Therefore, we could see that the basic metabolic rate should be 0.0002 unit/cycle.

### 3.2 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.

### 3.3 Architecture 2: Movement, measure lifetime as a function of speed



Figure 3: Error bar graph of varying speeds and directions

| Table … Robot Lifetime Mean and Standard Deviation for Different Speed Rate | | |
| --- | --- | --- |
| Speed Rate | Lifetime Mean(Cycles) | Lifetime Standard Deviation |
| 0.02 | Get data from Matlab |  |
| 0.04 |  |  |
| 0.06 |  |  |
| 0.08 |  |  |
| 0.10 |  |  |
| 0.12 |  |  |
| 0.14 |  |  |

Fig 3 where the robot eats everything it encounters. Where the error bars are the standard deviation for the robot at the given speed.

3.4 Architecture 3: Neuronal Classification of food

# 4. Discussion

# 5. Summary and Conclusion

# 6. Acknowledgements

# 7. References

1. Caudell, Thomas. "Flatworld."
2. Haykin, Simon S. *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice Hall, 1999. Print

# 8. Appendix A