Developing a model for location based route learning in a virtual world

BACHELOR THESIS, BY MARCEL ZUUR

AUGUST 11, 2013

UNIVERSITY OF TWENTE

FACULTY OF BEHAVIORAL SCIENCES, PSYCHOLOGY

DEPARTMENT OF COGNITIVE PSYCHOLOGY & ERGONOMICS

FIRST SUPERVISOR: PROF. DR. FRANK VAN DER VELDE

SECOND SUPERVISOR: DR. MATTHIJS NOORDZIJ

Abstract

A robot has been given a route learning task. It's goal is decision making based on the recognition of situations. It features a behavioural model which includes object recognition based on the ventral stream, dual-process decision making and motor control. This model tries to follow the computational cognitive neuroscience (CCN) ideals. Implementation is done using a combination of neural networks and programming.

PCA reveals that representations can emerge at different level of processing. Lesions study and PCA shows that location detection can be enhanced by combining vision and sonar. Results also show the benefits from using dual-processing decision making.

This thesis ends with stating that combining CCN modelling with traditional research can provide a powerful tool in understanding cognition.

Contents

1 Introduction **4**

2 Cognitive Computational Neuroscience **7**

2.1 Historical context 7

3 The robot, it's environment and the route learning task **12**

3.1 The robot and it's environment 12

3.2 Route learning task 15

3.3 Skills needed to complete the route learning task 15

3.4 Research question 16

4 Model for object recognition, decision making and motor control **17**

4.1 Ventral pathway for object recognition 17

4.2 Decision making: a dual process approach 21

4.3 Motor control 23

5 Implementing the model **24**

5.1 Materials 24

5.2 Neural Networks 25

5.3 Programming 32

6 Training **37**

6.1 Stage 1 learning objects and system 2 activation 37

6.2 Stage 2 training for action decision 39

6.3 Stage 3 testing the neural networks and the decision making model 42

6.4 Stage 4 training locations and sequence 43

7 Analysis **48**

7.1 Producing output 48

7.2 Analysis of the sonar detection network 50

7.3 Analysis of the location network 54

7.4 Sequence results 61

7.5 Activation of system 2 66

8 Conclusion and Discussion **68**

8.1 PCA Analysis 68

8.2 Lesion Studies 69

8.3 Model 69

8.4 Discussion 71

Bibliography **76**

A Leabra training algorithm **82**

B Programming code **84**

B.1 Detected objects 84

B.2 Creating the robot and the virtual world 85

B.3 Model implementation 85

1

Introduction

> Reductionist biology–examining individual brain parts, neural circuits and molecules–has brought us a long way, but it alone cannot explain the workings of the human brain, an information processor in our skull that is perhaps unparalleled anywhere in the universe. We must construct as well as reduce and build as well as dissect
>
> (Markram, 2012).

The person quoted here is Henry Markram director of the ”Human Brain Project” at the Swiss Federal Institute of Technology in Lausanne. He has the opinion that there is a need for a new paradigm in brain research, one that combines both analysis and synthesis. Whereas traditional brain research involves methods like dissecting, lesion studies, psychoactive drugs and the more modern imaging methods like EEG or fMRI, the ”Human Brain Project” has as it's ultimate goal a full simulation of a human brain. According to Markram (2012) this will run on supercomputers and incorporate all the data neuroscience has generated to date.

Although this sounds promising and should open up a whole new range of possibilities for brain research, it should be noted that this ambitious project is still in it's infancy and computing power is nowhere near what is required to create a fully functioning brain simulation. When looked at the rate computing power is increasing, brain simulation could be possible by the year 2023.

Besides the technical obstacles there are more problems lurking for such an ambitious project. A major one is is the problem of complexity. Djurfeldt, Ekeberg, and Lansner (2008) point out that with increased model complexity the uncertainty increases and in addition the

model loses explanatory power. Also increasing the number of parameters in a model means that more data is needed to determine those parameters. More data can be hard to come by and can also increase uncertainty. de Garis, Shuo, Goertzel, and Ruiting (2010) add that Markram's simulations do show similarities to real cortical dynamics, but that functional simulation has not yet been validated.

So a complete brain simulation faces quite a lot of obstacles and might be a bridge too far for now, but Markram's need for a new paradigm in brain research is shared by other scientists. Werbos (2009) raises the bar to a somewhat lower level than Markram when he states the following: "The most important challenge to scientific research today, in mathematical study of the mind, is to replicate and understand the level of general intelligence we can find in the smallest mouse." He continues with saying that we need to know how to build a mouse before we can build a man.

Cattell and Parker (2012) name three motivations why brain simulation can be useful:

1. Better understanding of how the brain works (and malfunctions) by creating simulations.
2. Ideas from simulations of neural networks may lead to the development of intelligent behaviours in computers.
3. Hardware architecture based on the massive parallelism and adaptability of the brain may lead to new computer architectures

The first motivation forms the basis for a relatively new field of research called cognitive computational neuroscience (CCN), a field that combines computational neuroscience, artificial intelligence, neural network theory and connectionism on the one hand and recent discoveries in psychology and neuroscience on the other. It's main focus lies on modelling the brain and or it's functions (Ashby & Helie, 2011).

The research that is presented in this thesis is the application of such a CCN model on a virtual robot operating in a three dimensional world. This robot will be given the task to learn and walk a route through it's environment. It will need to learn to recognize the different locations and situations in it's environment and decide which direction it should go based on what it recognises. This robot can be seen as a simplified version of a living creature and the world it operates in is purely designed for the task it is given.

This type of research is sometimes called the animat or "iguana" approach. This last term stems from a famous quote by Daniël Dennett.

> one does not want to get bogged down with technical problems in modelling the cognitive eccentricities of turtles if the point of the exercise is to uncover very general, very abstract principles. . . So why not then make up a whole cognitive creature, a Martian three-wheeled iguana,say, and an environmental niche for it to cope with? (Dennett, 1978)

Meyer (1996) summarizes best what this approach is about:

> The motto here is that it is possible to touch on issues of human intelligence according to a bottom-up approach which, originating in minimal architectures and simple environments, aims progressively to increase the complexity of these architectures and environments. If we take care to add to these architectures only those features which are necessary to the primary goals of perception, categorization, and the pursuit of autonomously generated tasks, it will become possible to resolve increasingly complex problems of survival without loosing the capacity to resolve the simplest.

This research will follow a similar bottom-up approach by first creating and testing all the structures needed to complete the route learning task. All these individual structures will be combined into a model that will control the robot's behaviour. The next chapter will give a more comprehensive explanation of CCN and it's ideals. For the development of the behavioural model, these CCN ideals shall be followed as much as possible.

The main research topic in this thesis is to show that when the scientific literature about object recognition, decision making and motor control are joint together in a CCN model, the underlying principles should help the robot operate in it's virtual world in much the same way as they do in humans or other animals.

Chapter 3 will discuss the robot, it's environment and the route learning task in more detail. After that the behavioural model and it's individual parts will be constructed. Different parts of the behavioural model need to be trained to make it possible for the robot to perform the route learning task. Performance of the robot, the behavioural model and it's individual structures will be analysed after performing the task. Finally the results will be discussed as well as the relevance of this type of research for science.

2

Cognitive Computational Neuroscience

The emerging field of Cognitive Computational Neuroscience (CCN) might be the field that is most in line with Markram's dream. This field tries to be a combination of traditional AI, connectionism, computational neuroscience on the one hand and psychology and neuroscience on the other. (Ashby & Helie, 2011). This chapter will provide some insight in this relatively new field. It starts with a brief history of the field, some challenges with CCN models will be discussed and finally four ideals for CCN models are presented.

2.1 Historical context

According to Ashby and Helie (2011) the field of CCN has partly come up because the vast majority of computational scientists are not psychologists and have no fundamental interest in behaviour and scientists in artificial intelligence are more interested in optimizing the performance of their models and not in modelling behaviour.

For a better understanding of the field of CCN it is wise to give a brief history of the fields of computational neuroscience and neural network theory and connectionism.

The term of computational neuroscience has come up somewhere in the mid 1980's although the origin of the field dates back some more decades. The mathematical model of the giant squid axon action potential by Hodgkin and Huxley (1952) is generally seen as the origin of computational neuroscience. This work led to the Hodgkin-Huxley model of the neuron, which is the cornerstone of the field. The Hodgkin-Huxley model is still the most widely used model for the modelling of single neurons. Computational neuroscience strives to be as biologically

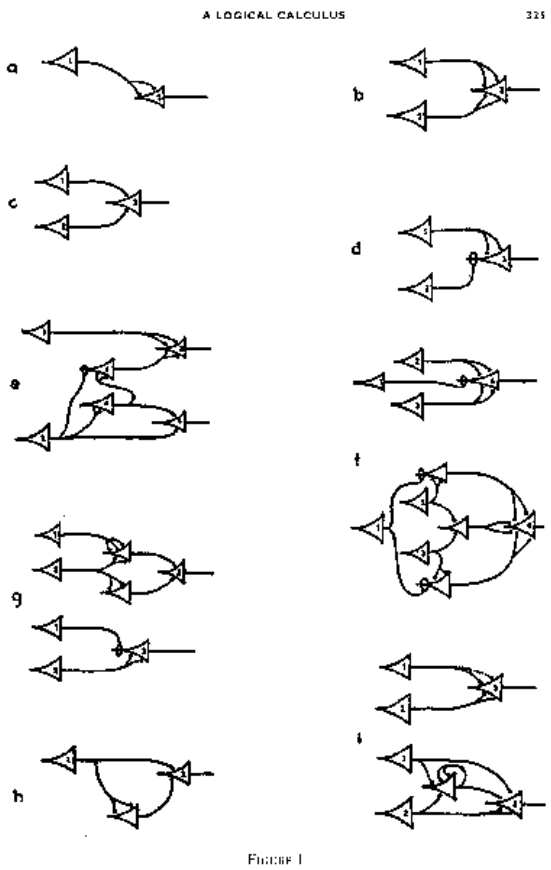

Figure 2.1. This network structure drawn by McCulloch and Pitts (1943) is seen as the first artificial neural network.

accurate as possible and most of their models include, at most, only a single neuron. Their models almost never accounts for behaviour, mainly due to the complexity of these single cell models (De Schutter, 2008; Ashby & Helie, 2011).

The origin of neural network theory and connectionism dates back to 1943 when McCulloch and Pitts (1943) created a model of artificial neurons. They describe a structure that later scientists would call a neural network , see figure 2.1. Hebb (1949) contributed with his famous ”Hebbian learning” a mechanism for unsupervised learning. More popularity came with the perceptron developed by Rosenblatt (1957) and the most popular algorithm for supervised learning, backpropagation (Werbos, 1974; Werbos, 1990; Buchanan, 2005; Benkő & Lányi, 2009).

The connectionists do see biological properties as an advantage, but do not see them as requirements. Neural networks have some features in common with the human brain, but the units in the neural network model typically do not behave like real neurons (Ashby & Helie, 2011). Besides that the most commonly used learning algorithm for neural networks

backpropagation is according to Crick (1989) unrealistic in almost every way.

The diverse areas in which neural networks are applied also show that researchers in artificial intelligence are not in general interested in modelling behaviour. Neural networks are used in robotics, but also in doing stock market predictions or predicting the hardness profiles of steel (Kim & Lewis, 1999; Zhang & Wu, 2009; Vermeulen, van der Wolk, de Weijer, & van der Zwaag, 1996).

The new field of CCN is then a combination of the tradition fields of computational neuroscience and connectionism, combined with new insights coming from psychology and neuroscience.

2.1.1 CCN Modelling Challenges

The main research method in CCN is computational modelling with neurobiological accuracy. Before going further on this topic first some issues with brain modelling in general.

When trying to model (parts of) the brain there are several things one should account for. Sejnowski, Koch, and Churchland (1988) define three different classes of brain models: First realistic brain models, second simplifying brain models and third technology for brain modelling. They warn that it is all too easy to make a complex model fit a limited subset of data. The also state that simplifying models are necessary to capture important principles, but are also dangerously seductive. A model can become an end in itself and lose touch with nature. They expect future brain models to incorporate the advantages of both realistic and simplifying models.

Djurfeldt et al. (2008) also warn for too much complexity. They state that more details from lower levels leads to more parameters, which in turn makes it harder to obtain a realistic model. A larger number of variables also makes studying the model much more difficult. According to them a model should have as few free parameters as possible.

Besides too much complexity there are more challenges for brain emulation. Cattell and Parker (2012) define the following major challenges:

- Neural complexity - Synapses can vary widely.
- Scale - Brain emulation requires massive computing power.
- Interconnectivity - Emulating in hardware is a massive ”wiring” problem
- plasticity - Synapses must be ”plastic”.

- Power consumption.

A good example of the power consumption problem is the ”Human Brain Project” this thesis started with. The extra-scale supercomputer which is needed for a full simulation will probably consume around 20 megawatts, which is the equivalent of the energy requirement of a small town in winter. This is quite a difference when compared to our brain which consumes around 20 watts (Markram, 2012).

2.1.2 CCN Ideals

According to Ashby and Helie (2011), what sets CCN models apart from other modelling traditions is that a model's validity is not only defined by it's goodness-of-fit to the behavioural data. For CCN models, this is just one criterion. CCN models have the extra constraint that the model should also function in a manner that is consistent with existing neuroscience data. It makes predictions about behavioural as well as neuroscience data and can be tested against both.

Ashby and Helie (2011) present four ideal principles for model building and testing in CCN:

1 The Neuroscience Ideal A CCN model should not make any assumptions that contradict current neuroscience literature. Four types of assumptions should be considered.

1. The model should only postulate connections among brain regions that have been verified in neuroanatomical trace studies.
2. Excitatory and inhibitory projections should be correctly specified
3. The qualitative behaviour of units in each brain region should agree with studies of single neurons in these regions.
4. Learning assumptions should agree with existing data on neural plasticity.

2 The Simplicity Ideal The neuroscience ideal doesn't mean that all neuroscience data should be incorporated into the model or that every feature of the model should be grounded in neuroscience. The simplicity ideal states that no extra neuroscientific detail should be added to the model unless there are data to test this component of the model or the model cannot function without this detail.

3 The Set-in-Stone Ideal This ideal states that after a constant is set in stone, it should not be considered a free parameter in any future application of the model.

4 The Goodness-of-Fit Ideal A CCN model should account for behavioural data and at least some neuroscience data.

These ideals try to incorporate the challenges stated in the previous section and should be helpful when building and evaluating CCN models. It is wise to note that Ashby and Helie (2011) mention that ideals should be seen as what they are, ideals and that no model can meet all the criteria.

3

The robot, it's environment and the route learning task

For the route learning task a virtual robot has been built using the Simbad robot simulator (http://simbad.sourceforge.net/). This is an open source framework written in the Java programming language which allows for the creation of simple robots and three dimensional environments. Robots can be equipped with different sensors. For instance they can be equipped with camera's, sonar or touch sensors. Worlds can be constructed using simple shapes like walls or spheres. Being more a framework than a finished product and it's open source nature makes it highly flexible. This allows Simbad to be easily integrated in other projects.

This chapter will discuss the robot and the environment that have been build with the help of Simbad. After that the route the robot has to learn will be discussed, together with the possible obstacles the robot has to overcome to complete the task.

3.1 The robot and it's environment

The world the robot is placed in is made up out of straight walls. These walls form hallways which are connected to each other. Together they form a maze-like structure. The walls all have different colors so that every location in the world can be recognised on it's unique color pattern. Each location which has more than one direction to choose from has been assigned a letter. The complete world can be seen in figure 3.1.

The robot can move in a forward direction through it's environment. It has the abilities to

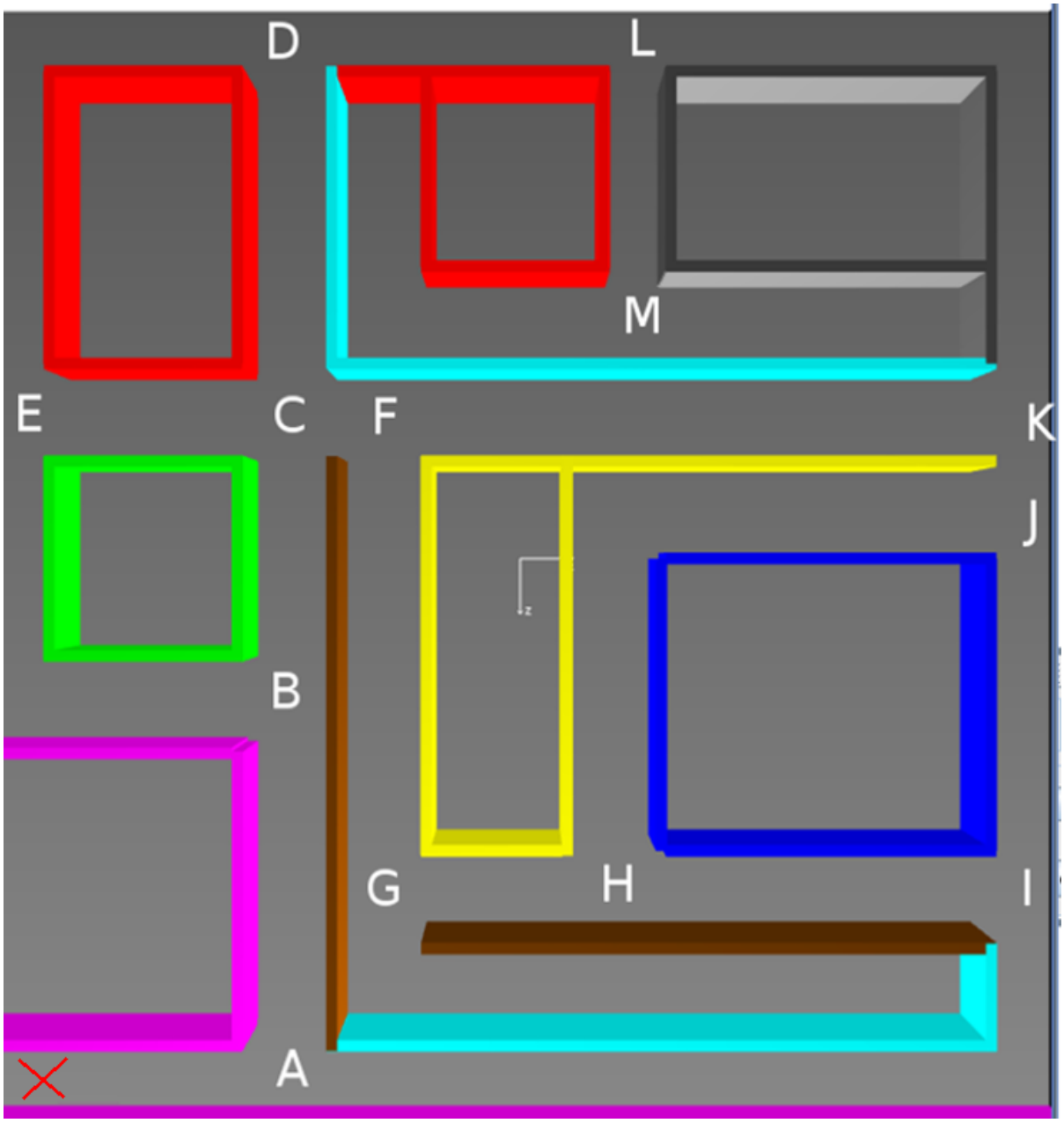

Figure 3.1. The world the robot operates in. The letters indicate locations which have more than one option for the robot to choose from. The red X at the lower left corner indicates the starting position of the robot.

stop, take a left turn, take a right turn and turn around. The robot is equipped with two types of sensors to give it awareness of it's surroundings. The first type of sensor is a sonar belt. This belt contains 24 sonar sensors that are placed at an equal distance around the robot, giving it a 360°view of it's surroundings. Because the sonar sensors are placed in a belt, they are all located at the same height. Figure 3.2 shows how these sonar sensors are placed around the robot.

A sound signal is send out in a straight beam and bounces back to the robot when an object

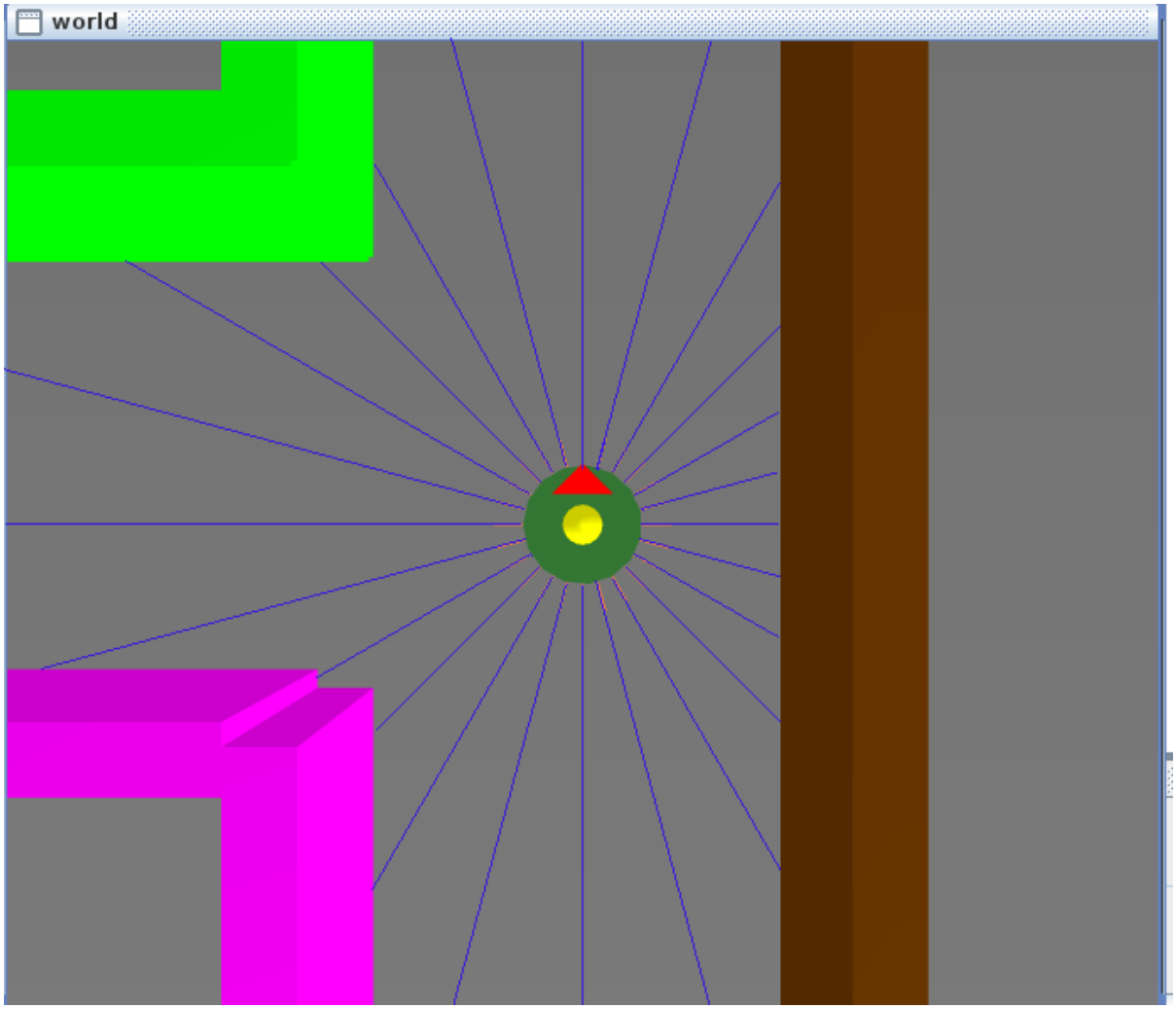

Figure 3.2. All the 24 sonar sensors are placed at equal distance around the robot, providing a 360°image of its surroundings.

is close enough. The strength of the returning signal depends on how close the object the beam reflected from is located. Signal strength decreases when an object is further away and even becomes zero when the distance becomes too large. The returning signal then gives two types of information. First the 24 sonar sensors form a one dimensional structure in which every sensor can be seen as a dot. Activation of one or more of these dot's mean that a object is present at that given location. Second the strength of the signal represents the distance between the robot and the object detected.

The second type of sensor provides color vision by using three camera's. The first camera is located at the front side of the robot. The others are at the left and right side of the robot, making 90°angles with the front camera. Each camera has a 90°viewing angle, so the three camera's cover a 270°angle. The sonar sensors are used to recognize shapes and color vision is used to recognise locations which cannot be recognised using shape information alone.

Table 3.1
The Route the Robot has to Learn.

	Location	Decision		Location	Decision
1	A	Forward	12	D	Right
2	I	Left	13	L	Right
3	H	Forward	14	M	Right
4	G	Left	15	M	Left
5	G	Forward	16	L	Right
6	F	Left	17	K	Forward
7	C	Right	18	J	Right
8	D	Left	19	H	Left
9	E	Forward	20	I	Right
10	B	Left	21	A	Forward
11	C	Forward			

3.2 Route learning task

Since the ultimate goal for the robot is to learn a complex route based on the locations it recognises, it was chosen to create a rather complex route that visits each location a least one time. The locations that the robot has to learn are the places that have been assigned a letter. These locations have more than one direction to choose from, so the robot needs to make a directional decision there.

Some locations are visited two times, either coming from different directions or the same direction, but with a different decision. Table 3.1 shows the route the robot has to learn. It starts at location A and will eventually return at the same location. The route has several possible difficulties. For instance when visiting G for the first time it has to take a left turn, which will lead the robot to a dead end. The robot has to come back to G and then cross straight. This means that is has to learn location G from two sides and take two different decisions. Other possible obstacles are that locations F and C follow each other rapidly and the small passage when taking a right turn at L.

3.3 Skills needed to complete the route learning task

The robot will need several skills to accomplish the route learning task. First it needs the ability to walk through the world in a reliable manner. For that it needs a system that can detect places/situations in the world which need decision making. It needs to know that when it has arrived at a left turn for example it now has to change it's direction and go left. This

type of decision making is still relatively easy because there are no options to choose from. It becomes more complicated when arriving at a crossing where it needs to decide between multiple directions. When a decision has been made, this decision then has to be transformed into an action.

Recognizing, deciding and then performing an action is not enough for moving through the world in a reliable manner. The thing missing here is knowing when to perform an action. For example you should only turn left when there is enough room to make a left turn. So a complete system needs to: First recognise a location. Second, know what options are available. Third, if necessary decide between the options. Fourth select an action. Fifth know when to perform that action. Sixth perform that action at the right moment.

These six steps should give the robot the ability to walk through the world, but for route learning the picture is not complete yet. Route learning is done by recognising the current location and knowing which direction it should go at that location. Although it may look that way at first, it is not enough for the robot to learn direct associations between a location and the direction the robot must go. For more realistic route learning it is necessary that locations can be visited more than one time, either from the same or from another direction and that the action to be taken at that location can differ from the one chosen at a previous visit. Route learning therefore cannot be based on direct association, but rather on sequence learning. With sequence learning, a decision is based on all the previous locations and decisions in a sequence. This means that for example visiting location A for the first time differs from visiting that location a second time, because their history of previous locations and decisions is different.

3.4 Research question

The main research question here is how can current knowledge about brain structures be combined to create a model with which the robot can complete the route learning task? For reasons that will be explained later it is hypothesised that for object recognition the robot can benefit from current knowledge about the human ventral stream. Decision making should benefit from incorporating a dual-process model of decision making and knowledge about structures involved in human motor control should be helpful for controlling the robot's actions.

The next chapter will discuss the the model that will control the robot and the scientific literature at which all the different parts of the model are based.

4

Model for object recognition, decision making and motor control

A model will be constructed that will control the actions of the robot. Because this research tries to conform to the CCN ideals, this model will be based on structures that are known from scientific literature. The final model as it is used in this thesis is shown in figure 4.1. It consists of structures for object recognition, decision making and motor control. How the model is implemented will be discussed in the following chapters, this chapter focuses on the different theories that have led to the model.

4.1 Ventral pathway for object recognition

Although the robot uses sonar for object recognition instead of vision it is hypothesised that object recognition with sonar can be achieved using a system that is modelled after the human ventral stream. Sonar is used by different animals like bats for object recognition, but there are also some human examples that have developed some form of sonar for that purpose. These humans were blind from or early after birth and learned themselves to produce sounds that reflect from objects in their presence. The brain captures the returning sounds and can use them to create images of the objects in their environment. Study also shows that they use the same ventral stream areas in the brain normally used for the processing of vision (Thaler, Arnott, & Goodale, 2011). This is a principle called neuroplasticity. This process is also documented in blind people who have enhanced hearing capabilities because occipital areas normally used for

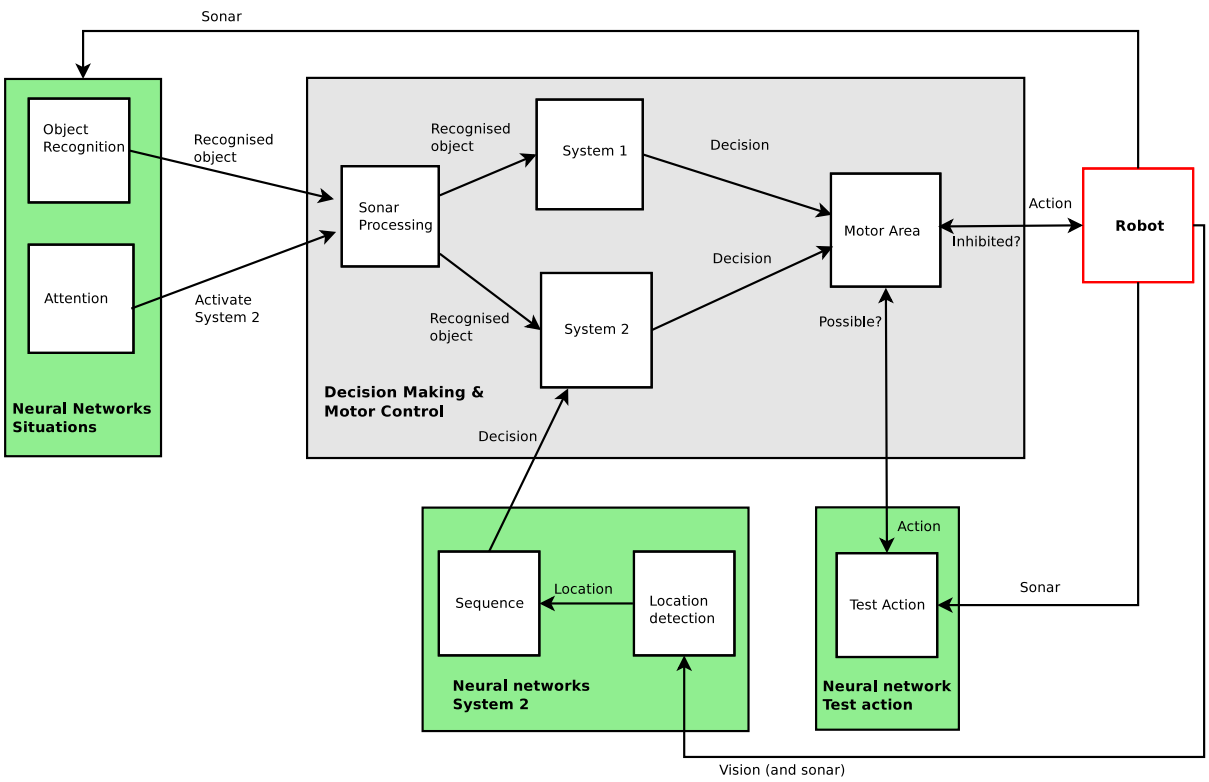

Figure 4.1. The object recognition, decision making and motor control model

the processing of visual stimuli are substituted for auditory processing (Collignon, Vandewalle, et al., 2011; Collignon, Lassonde, Lepore, Bastien, & Veraart, 2007). The areas in the ventral stream might then be organised in such a way that it is not bound to visual information, but can be used to process different kinds of information.

Another reason to choose for the ventral stream is that the type of information that needs to be processed when using sonar, has some similarities with visual information. The picture coming from the sonar sensors is a two-dimensional image with on the one axis it's horizontal position in space and on the other the distance between the object and the robot. Figure 4.2 shows how the sonar signal picked up at a front left situation, as can be seen in figure 3.2, can be translated into a two-dimensional image.

Sonar signals coming from an object in the environment can then be seen as two dimensional pictures. Simple geometric shapes in these pictures can be used to discriminate between the different objects. Objects do not change in shape, but the picture can change in size or shape depending on the distance and the angle between the robot and the detected object. This is much like vision where objects themselves do not change, but their picture can be different depending on the location of the observer in relation to the object. What is needed then is a structure that can recognise objects from different angles and distances. It needs the capabilities

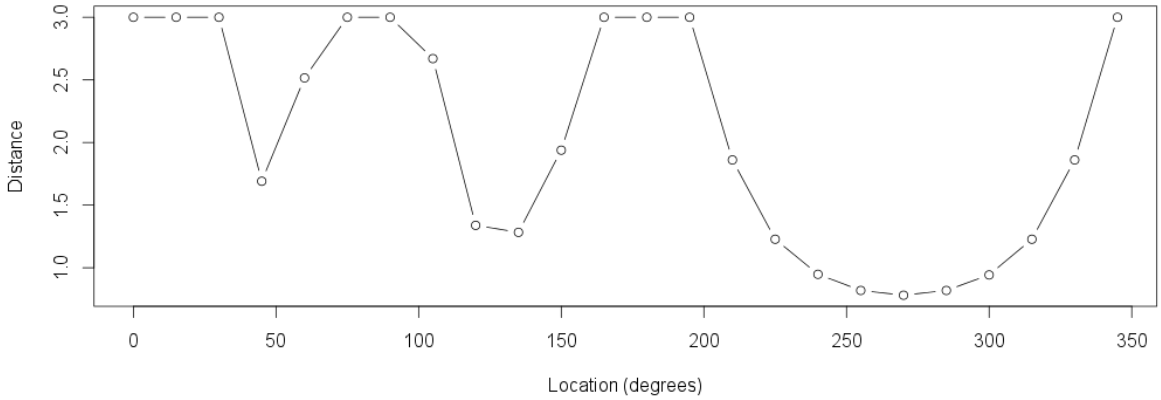

Figure 4.2. Front left situation sonar signal translated into a two dimensional image. X-axis shows the horizontal position in space expressed in degrees, starting at 0 degrees at the front of the robot and increasing 15°counter-clockwise. The y-axis shows the distance between the robot and the object. A distance of three is the maximum distance the robot can detect.

to use shape information to detect objects in a spatial invariant way and this is exactly what the ventral stream is capable of doing.

4.1.1 Ventral Stream

The ventral stream is associated with object recognition and travels trough different brain areas. It starts in the primary visual cortex V1, passes through the secondary visual cortex V2 to the V4 area and ends in areas in the inferior temporal lobe IT. This stream is not just a feed forward system, but strong feedback connections exist between the areas.(Lamme, Supèr, & Spekreijse, 1998; Ungerleider & Haxby, 1994; Mishkin, Ungerleider, & Macko, 1983).

Although anatomical studies show proof of the existence of such a pathway in the brain and that the ventral stream is associated with object recognition, much less is known about how these connected brain areas make us recognise objects. Evidence does show that receptive fields (RF) start small in the V1 area, where neurons only code for a very small part of the image and increases with the passing of each area. All the way through the IT areas where object representation becomes the most abstract and spatial invariant (see figure 4.3.

From all the areas in the ventral stream, the primary visual cortex V1 is the most studied area. Hubel and Wiesel (1998) did a lot of research on cells in the visual cortex of cats and monkeys and discovered that cells in that area have RFs that respond strongly to bar or edge-shaped patterns. This area and the following areas except for IT layer are organised in a retinotopic

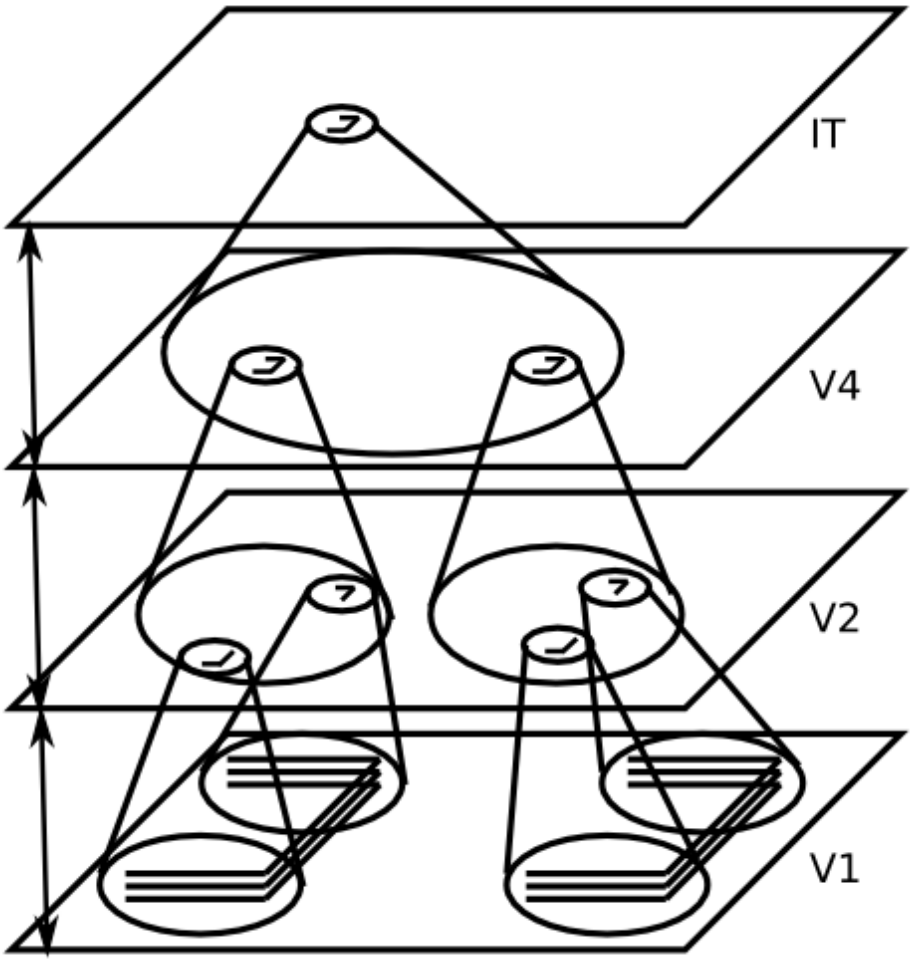

Figure 4.3. The receptive field of the neurons start very small in the V1 layer, but the size increases with the passing of each layer. From the detection of simple bars to complete abstract and spatial invariant representations of objects.

fashion (Oram & Perrett, 1994).

The V1 is then connected to the V2 area. The neurons in this area show much similarity to the neurons in the V1 in the sense that they respond to simple shapes and colors. The increased size of the receptive field makes it possible that V2 neurons can respond to somewhat more complex patterns. This area seems to play an important role in the analysis of contours and textures (Anzai, Peng, & Van Essen, 2007)

Much less is known about the function of the V4 and IT areas. Research done by Gross, Rocha-Miranda, Bender, et al. (1972) already shows firing activity of neurons in the IT area when specific classes of objects are presented. This area also contains the fusiform face area in humans which is recognised as being specialised in face detection. Not much is known about how the V4 and IT area represents objects and how they are organised. The V4 area and IT area both are linked to the representation of form and colour. The major difference between them is

that the V4 layer still has the retinotopic organisation, but loses this when the transition to the IT areas is made. This means that represented objects become completely spatial invariant. This loss of retinotopic organisation might be due to the increasing receptive field. The receptive fields increase with a factor 2-2.5 with the passing of each layer. (Oram & Perrett, 1994; Tanaka, 1996; van der Velde & de Kamps, 2001) It is also hypothesised that organisation in the temporal lobe occurs closely to related areas. So is the fusiform area located close to the amygdala and the reason could be that facial expressions and the recognition of emotions are closely related. In the same way objects that are more related to movement could be represented more closely to the motor areas (Mahon & Caramazza, 2011; Rust & DiCarlo, 2010; DiCarlo & Cox, 2007; Peissig & Tarr, 2007).

4.2 Decision making: a dual process approach

When the robot is capable of recognising objects, the next step is to let it take decisions based on what it has detected. For that it needs a decision making system. There are two types of situations in the world of the robot that require a form of decision making. The first is the situation where only one direction is possible. For instance when the robot walks into a hallway with a dead-end the only option is to turn around and go back. In other situations, like a crossing, there are more options to choose from. In situations which have only one possible option, decision making can be done in a habit like form. When there are more options available a more elaborate process of decision making is necessary.

This type of distinction between two decision making systems forms the basis of many theories of ”human” behavour. Evans (2008) wrote a review about several of these so called dual-process theories. According to Evans (2008) almost all authors agree on a the distinction between a system (system 1) that is unconscious, rapid, autonomous and has a high capacity while the other system (system 2) is conscious, slow and deliberative.

It was long thought that system 2 is uniquely human en has evolved much later than system 1. This idea stems mainly from the association system 2 has with human processes like language or the ability to perform cognitive acts that are beyond the capabilities of animals. Nowadays more evidence is showing that these systems might also be available in animals (Evans, n.d.; Evans, 2003). Research by Toates (2006) even shows evidence for a distinction between an associative and higher order system control processes in many higher animals and mentions that dual control appears to be an adaptive solution for the control of behaviour.

4.2. Decision making: a dual process approach

Because so much evidence in human and animal research point to dual modes of processing, it is expected that the robot can benefit from incorporating such a theory in the model.

The dual process model that will be used in this research is an adaptation of the ”Reflective and Impulsive model” by Strack and Deutsch (2004). In their article the authors use ten statements to sum up their model. Although they use their model to explain human social behaviour, this model can easily be adjusted to the situation of the robot which doesn’t have any social interaction. The first six statements are in a somewhat adjusted form used here and form the basis of the robot’s decision making model.

1. Behaviour is the effect of the operation of two distinct systems of information processing: a reflective (System 1) and an impulsive (System 2) system.

2. Both systems operate in parallel. The impulsive system is always engaged in processing whereas the reflective may be disengaged.

3. The reflective system requires a high amount of cognitive capacity whereas the impulsive system requires little cognitive capacity.

4. Elements in the two systems are connected by different types of relations. In the reflective system, elements are connected through semantic relations to which a truth value is assigned. In the impulsive system, the relations are associative links between elements.

5. There is a final common pathway to overt behaviour in the impulsive system, that may be activated by input from the reflective and the impulsive system.

6. The systems uses different operations to elicit behaviour. In the reflective system behaviour is a consequence of a decision and in the impulsive system behaviour is elicited through spreading activation.

The second point poses somewhat of a problem, because how does the system know how to activate system 2? Strack and Deutsch (2004) mention that activation of the reflective system depends on the intensity of a stimulus and how much attention it receives. The model will have to contain some sort of stimulus driven attention system that discovers objects that need system 2 for further processing. This kind of exogenous attention needs a bottom-up mechanism that can detect these kinds of objects as early as possible in the processing. Attention can then activate the areas than can perform the system 2 processing (Theeuwes, 2010; Corbetta, Patel, & Shulman, 2008).

4.3 Motor control

When the decision making system has reached a final decision, this outcome can then be transformed into an action. To achieve this, the model has to be extended with a motor control system. This part of the model is roughly based on some basic structures that perform motor control in the human brain. The first structure is the primary motor cortex which can execute a desired action. The primary motor cortex is connected to the muscles.

Before an action gets executed another structure, the pre-frontal cortex will predict the outcome of that action. Only when the desired outcome can be reached the action will be performed. In humans another structure, the basal ganglia, determines which movement gets selected by stopping to inhibit it, making sure there are no unwanted movements (Kalat, 2007). This model will incorporate the basic functionality of these structures by creating a motor system that will first try the action that is the outcome of the decision making system and will only try to execute that action when a desirable outcome is predicted. An inhibition system will make sure that only one action will be executed at a time.

5

Implementing the model

The implementation of the robot's behavioural model consists of a mixture of different neural networks which communicate with Simbad. This chapter will first discuss which tools are used to build and connect the different neural networks. The rest of the chapter will discuss in detail how the different structures were made.

5.1 Materials

The proposed model will be implemented using different software tools. The ventral stream is modelled in a neural network simulator called Emergent, the decision making part is programmed in Java and the motor control part is using a combination of Java programming and neural networks. The programming is needed to ”glue” all these networks together. It does that by providing two-way communication between the robot and the neural networks.

For the creation of the different neural networks used in this research, the Emergent neural network simulator will be used (Aisa, Mingus, & O'Reilly, 2008). This piece of software allows for the creation of neural networks, ranging from simple basic networks to very complex ones. It also comes with different training algorithms available, including back-propagation and Leabra. Emergent was chosen for this research because of a number of reasons. First it makes it possible to create very complex networks. Networks can have multiple layers, thousands of neurons and complex connectivity. Second Emergent is equipped with it's own scripting language with which you can completely control all of Emergents features. It already has some good default programs available written in that scripting language that can easily be adjusted to

personal needs. For example you can create a program to start a training from a specific training set, save the weights and record activation of specific neurons to a data set.

A third reason is that emergent can act as a server. This means that when server functionality is activated, external sources can create a connection with Emergent. This connection then allows for remote reading from and writing to datasets and the execution of programs, including the ones you have created with the scripting language.

Fourth is the Leabra training algorithm. This training algorithm claims to be more biologically plausible than the more common backpropagation algorithm (Petrov, Jilk, & O'Reilly, 2010) and is therefore more suited for CCN modelling. See appendix A for a description of this algorithm.

5.2 Neural Networks

For the implementation of the model it is necessary to develop several neural networks. It starts with object detection then an early attention mechanism for the activation of system 2. This is followed by a structure that can detected right moments for action. After that a structure for location detection will be developed and finally a neural network will be built that can learn sequences.

5.2.1 Detecting situations

As stated in the previous chapter the robot uses sonar information instead of visual information. This choice is also a practical one. The richer sonar signal allows for much less input information. For stereoscopic sight using two very low-resolution images of 20x20 pixels would already require 800 input neurons. Images with such low resolutions would probably be insufficient for object recognition. Reliable object recognition would probably require many more neurons, which comes at the cost of needing too much computing power.

Because of the richness of the sonar data, just the 24 sonar sensors are enough to gather information that is sufficient for object recognition. Each sonar sensor is connected to one input neuron of the network. The amount of activation of an input neuron corresponds to the strength of the sonar signal coming from the sonar sensor this neuron is attached to.

The complete network consists of six layers, input, S1,S2,S4,IT and output. The complete structure can be seen in figure 5.2. Since the letter 'V' in V1, V2 etc stands for vision, this letter

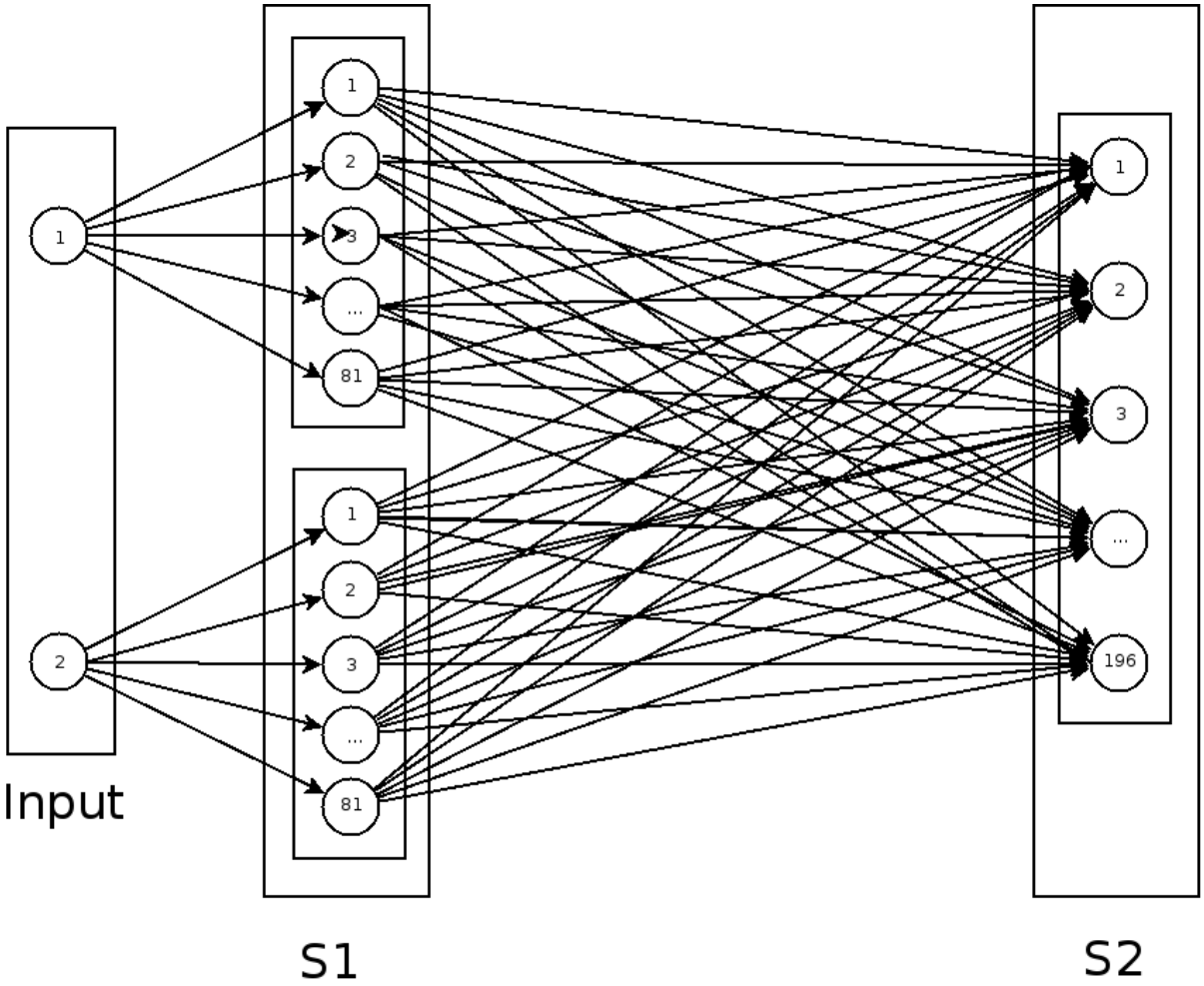

Figure 5.1. An example of how the input neurons are connected to groups of neurons in the S1 layer. The first input neuron is connected to the first group of 81 neurons and the second input neuron is connected to the second group of neurons. Both groups are then connected to a single group of 196 neurons in the S2 layer. In the same way, the 24 input neurons from the sonar network are connected to 24 neuron groups in the S1 layer. Each of the six groups in the S2 layer are then connected to four of the neuron groups in the S1 layer. The S4 which has two of these groups are connected to three of the neuron groups in the S2 layer.

is replaced by the 'S'.

The S1 layer is the first layer of processing and is divided into 24 groups of neurons which all have 9x9 neurons in it. Each input neuron is connected to one of these groups. This means that the activation coming from a single input neuron is processed by 81 neurons in the S1 layer. The S1 layer then has a total of 1944 neurons. Figure 5.1 gives an example of how the layers are divided in groups of neurons. Since each neuron in one of the neuron groups in the S1 layer is only connected to one input neuron, the RF in this layer is the smallest. Each neuron codes only for a small fraction of the sonar image. This type of connectivity between the input layer and the S1 layer also means that processing is organised in a retinotopic fashion.

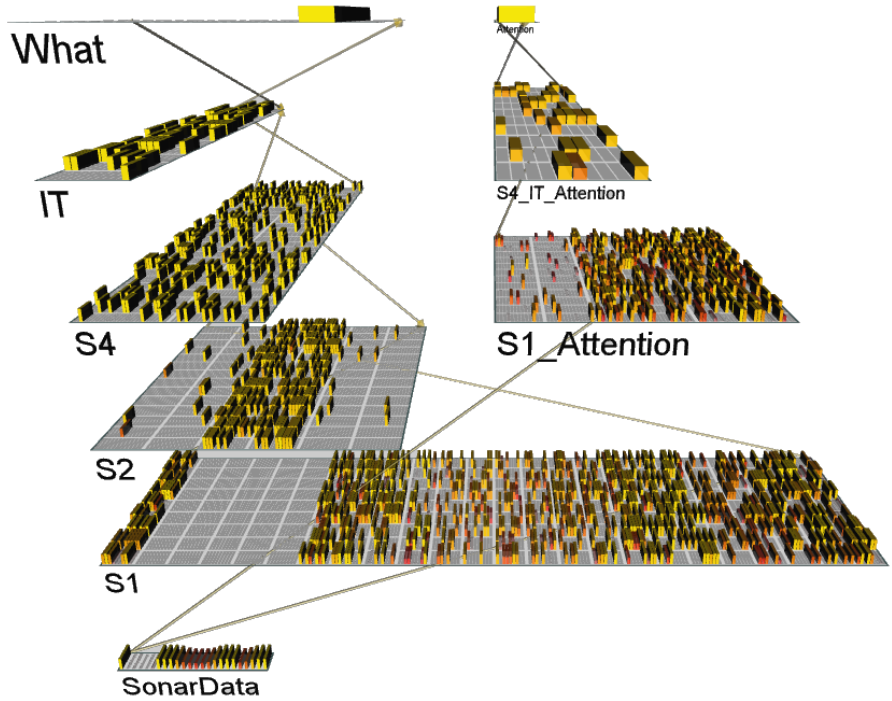

Figure 5.2. The neural network model used for object recognition and attention. The S1, S2, S4, IT and Output layers are modelled after the ventral stream. The S1 attention, S4 attention and Attention layers use a simplified version of this model to detect situations which need system 2 activation.

The S1 layer then projects onto the S2 layer. Since the RFs increase while passing through each area of the ventral stream, the RF of neurons in the S2 layer should be bigger than in the S1 layer. This is done by having only 6 neuron groups instead of 24. These groups now have 196 neurons in it. The total number of neurons in this layer is 1176. The RF gets four times bigger because the first 4 groups of neurons in the S1 layer are connected to the first group in the S2 layer. Groups 5-8 are then connected to the second S2 group of neurons etc. The number of neurons within a neuron group increases to allow more advanced processing. Besides the feed-forward S1-S2 connection a feedback connection is also made. In the feedback connection a full projection is used. Retinotopic organisation is still present. For example input coming from the left side of the robot is processed on the left side of the S1 layer and is also processed on the left side of the S2 layer.

The transition from S2 to S4 has a similar construction. RF increases, because the S4 layer contains just 2 groups with each have 676 neurons in it, giving it a total of 1352 neurons. A feedback connection is also made in the same way as between the S2 and S1 layer. Because there is still a division in neuron groups, organization is still in a retinotopic fashion.

The IT as endpoint of the ventral stream has the largest RF. It has just one group consisting of 144 neurons. The retinotopic organisation ends here with the full connectivity between the S4 and IT layer.

Last the IT layer is connected to the output layer where the objects are represented as single neurons. The connection between these two layers is a full feed forward and feedback projection.

5.2.2 Switching between System 1 and System 2

A different neural network will use the same sonar input information as the network described above to detect situations where system 2 needs to be activated. This can be seen as an early attention mechanism and hence the built structure is called attention.

This structure is a somewhat simplified version of the ventral stream model. The sonar input is projected on the attention S1 layer. This layer is just like the S1 layer the first area of processing, including the feedback projections. The difference is in the size of the receptive field. Detecting situations that need attention require less detail processing than object recognition. The S1 attention layer has 6 groups of neurons consisting of 169 neurons. The first 4 input neurons are then connected to the first S1 group. The second layer of processing is called the S4 IT attention layer, and is made up out of a single group of neurons consisting of 100 neurons. The output layer contains just a single neuron which indicates whether the robot is in a high or low attention situation. Activation of this neuron indicates the need for system 2 processing.

Because it makes use of the same input information as the decision recognising network it is directly attached to it as can be seen in figure 5.2. Although they share the same input information and are activated at the same time, there is no other connection between them, which means they operate completely independent.

5.2.3 Determining right moment for action

Another import structure is the structure that determines whether it is the right moment to perform an action. By action is meant all the possible directions, turning left, turning right or crossing straight. This structure is unaware of the action that is decided. It merely detects whether one or more of the three possible actions is possible. A positive result from this network can be seen as a green light for the motor system to tell the robot it may perform an action.

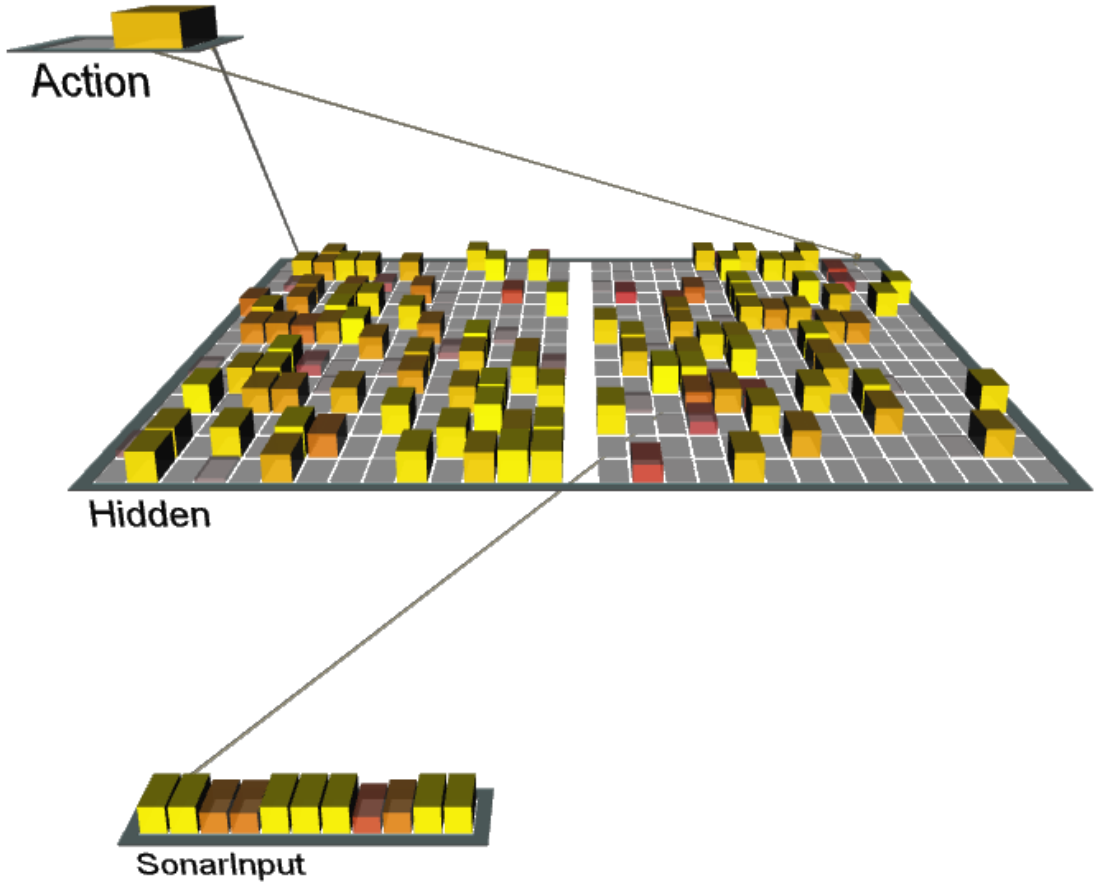

Figure 5.3. Neural network that detects whether performing an action is possible. This is a three layered structure with feedback connections between the layers and the hidden layer is divided into two neuron groups. The right output neuron is activated when it is a good moment to perform an action, otherwise the left output neuron will be activated.

Compared to the other networks used in this research this network is a somewhat simpler model. It has an input layer which contains 24 sonar input neurons, a hidden layer with 2 groups of neurons each having 196 neurons and an output layer with just two neurons. The two neurons code for an action being possible or not. The result from the output layer is fed back into the hidden layer to enhance the learning effect. Figure 5.3 shows the network.

5.2.4 Recognizing Locations

This network comes in two variants. One that is solely based on vision information and one that also uses sonar information. The idea behind this is that vision information should be enough on it's own for recognizing locations, but that the shape information coming from the sonar sensors

might enhance the detection of locations. These networks are based on the same principles as the network from section 5.2.1. First described here is the vision part.

A major difference with the network from section 5.2.1 is that there is no V2 layer. Because the major discriminating factor here is color information, shape information should be less relevant, therefore the V1 and V2 layer are combined to just one V1 layer.

Just like the network from 5.2.1 the transition from one layer to the next means the neurons RF will increase. There are three input layers each containing 16x48 neurons. These input layers are the red, green and blue components of vision input. This input is the combined input of the three cameras that are attached to the robot. These cameras create images which have a size of 16x16 pixels. The three layers then come together in the V1 layer. The input layers are divided into groups of 4x4 neurons, which are connected to neuron groups in the V1 layer. The V1 layer has 12x4 of these groups having each 296 neurons.

RF then further increases going to the V4 layer which only consists of three unit groups with 28x28 neurons, and further increases with the connection from V4 to IT. All the three unit groups from V4 are connected to the IT layer which is made up out of 9x9 neurons. The IT layer is then connected to the output layer consisting of 13 neurons, corresponding to the 13 different locations in the test world. Besides the feed forward projections each layer has a feedback projection to it's previous layer. See figure 5.4 for the complete structure.

This network differs from the sonar network from section 5.2.1 in that a little Gaussian noise was added to the activation of the neurons. The idea behind this is that this might enhance learning. Because of lightning effects in the world a color doesn't always look the same, it can be brighter or more shaded depending on the robot's position. Just a small amount of noise with a variance of 0.0005 was added to compensate for that. Different variances were tried, but this number proved to give the best results.

5.2.5 Extended with sonar

A second version of this network was made that also uses sonar information. The vision part is exactly the same, but some layers were added to provide the sonar processing. Input consists of the input from the 24 sonar sensors. Input goes to the S1 layer which consists of 6 unit groups which all consist of 14x14 neurons. RF also increases here travelling from the S1 to the S4 layer, which has 2 unit groups consisting of 18x18 neurons. These two S4 unit groups are then connected to the same IT layer as where the V4 layer is connected to. Figure 5.5 shows the

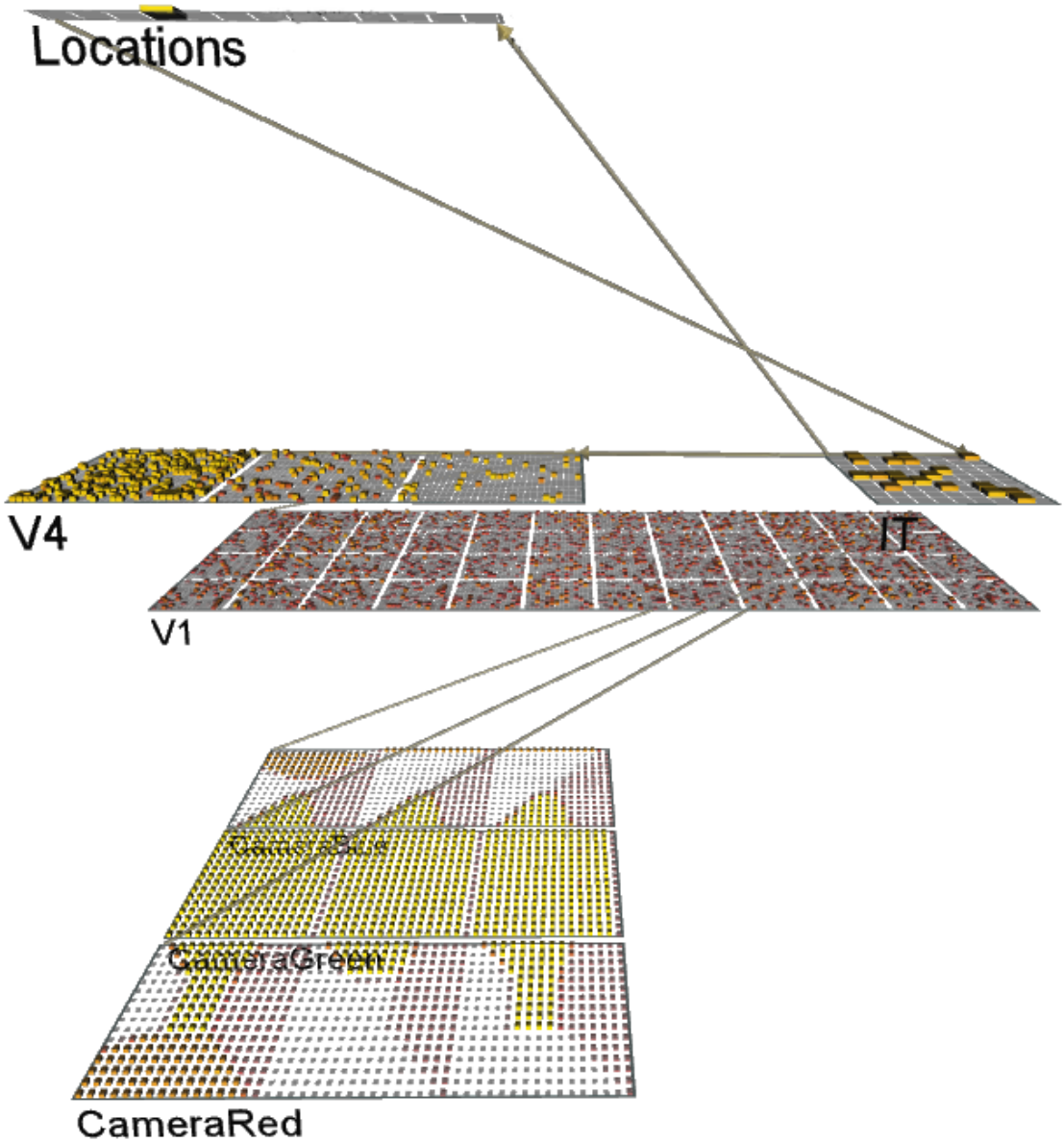

Figure 5.4. Neural network that detects locations based on vision information. This structure is also modelled after the ventral stream, the only difference is that there is no V2 layer.

location detection network enhanced with sonar processing.

5.2.6 Remembering Routes

Remembering the route is done with the well known Elman network (Elman, 1990). This is a type of recursive neural network that can remember sequences. The 13 different locations are used as input and output is one of the three decision left, forward or right. The input layer is connected to the hidden layer. The hidden layer is connected to the output layer, but also recursively connected to a context layer. This provides the ”memory” that can remember the

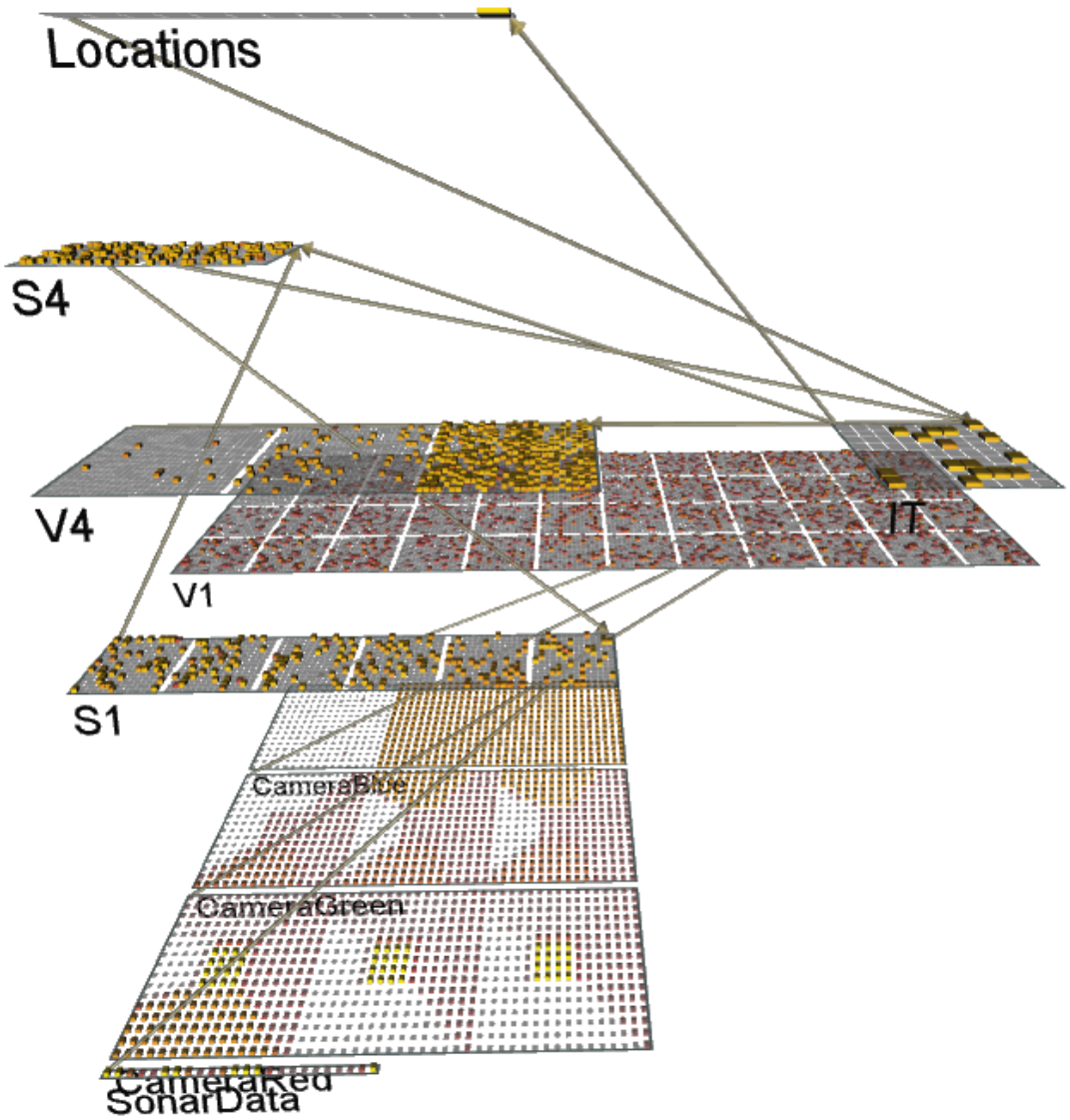

Figure 5.5. This is the same neural network for the detection of locations, but then extended with sonar processing capabilities. The sonar part, the SonarData, S1 and S4 layers, is also modelled after the ventral stream. Vision and sonar come together in the IT which is then connected to the output neurons.

sequence.

5.3 Programming

Programming is needed to let all the different parts work together. The structures that were programmed are the sonar and vision modality, system 1 and system 2 and the motor system. The programming language used here is Java.

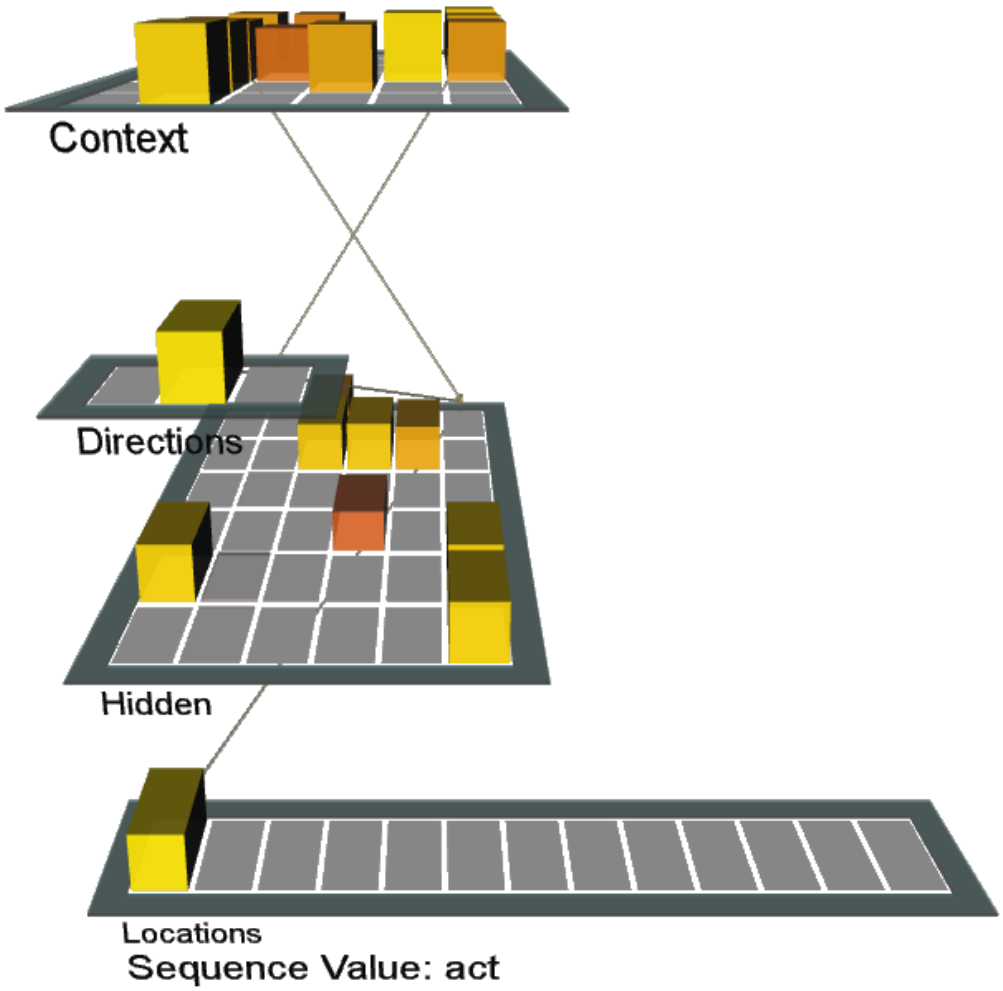

Figure 5.6. The recursive Elman network used for sequence learning. An input layer is connected to a hidden layer. This hidden layer is connected to an output layer, but is also recursively connected to a context layer.

5.3.1 Sonar modality

The main goal of the sonar modality class is to provide a way to let the robot communicate to the two neural networks that use sonar information and then retrieving back the outcomes of these networks. It will retrieve all the information coming from the 24 sensors and transform it into a format that Emergent can use as input. It can then call the Emergent program that will run the network and when that is finished it will collect the result the two sonar networks have produced. The first result is the result coming from network that detects the decision situations. The output from emergent is then parsed and transformed in a java object called an Enum. Listing 1 in appendix B shows that structure for all the possible results. Internally all output coming from Emergent is transformed into such Enum objects. The second result is from the

network that decides whether system 2 should be activated or not. Since this network only has one neuron acting as a boolean value, activation from this network will set a value to true so that the rest of the system can know that system 2 has to be activated.

5.3.2 Vision modality

The vision modality structure shares much characteristics with the sonar modality structure, but as the name implies sends vision information coming from the robots three cameras. The images coming from the three cameras are first combined into a single image. The cameras are full color cameras with a resolution of 16x16 pixels, so the combined image consists of 48x16 pixels.

Since the goal is to let the robot learn locations based on color information it was decided to split the combined image up into three separate images. The first image consists of the amount of red in the picture, the second the amount of green and the third the amount of blue in the picture. This is much in line with human vision which also uses different three different types of color receptors in the eye. This of course does increase the amount of input by a factor three. Each pixel value is directly translated into an input neuron value, meaning that input for the neural networks consists of 2304 input neurons.

5.3.3 System 1

As described in section 4.2 system 1 will always be active and produce it's own decision based purely on association. Based on this last fact it was chosen to completely write this in java instead of creating a neural network for this. Although neural networks are perfect for learning direct associations the end result would have been exactly the same as directly coding these associations. Listing 4 shows that the java implementation of system 1 is not much more than a switch of an Enum structure that connects the object recognised by the neural network described from section 5.2.1 to a decision. For example a left turn is directly associated with the decision to go left.

5.3.4 System 2

System 2 will use the same input as system 1 for it's decision making, but can make more complicated decisions. The way the java code is set up, it is possible to use different implementations of system 2 and hence have different ways of coming to decisions. The version

of system 2 that will be used to let the robot walk a learned route will consist of first activating the location recognizing network through the vision modality and then send the recognized location to the sequence network to reach a decision.

Other implementations of system 2 were used during the learning process for training and testing purposes. For instance one version of system 2 was used to create training sets for the location recognising network. Another example is an implementation of system 2 that makes random decisions. This last one proved handy for testing how long the robot could move around in the world without crashing.

5.3.5 Motor system

The motor system is the place where the system 1 and system 2 decisions come together. This system was built in such a way that a system 2 decision is always favoured above a system 1 decision. It is programmed in such a way that it will first look for a system 2 decision and only when there is no system 2 decision it will try to perform the system 1 decision.

Deciding between system 1 and system 2 is not the only task to perform. When it has decided between the two systems that decision has to be transformed into an action that the robot actually has to perform. The robot has to determine whether the action that comes out of the decision making process can actually be performed at that moment. It is entirely possible that a left turn is detected, the desired action is to go left, but the robot still has to move some distance for the left turn to be possible. To prevent all this the network that detects the possibility of an action as described in section 5.2.3 will be called to test whether the desired action is possible. Then and only then will the robot be given the green light to perform the action.

This way of preventing action also has another consequence. The decision making process is an ongoing process, which means that while performing a requested action is inhibited, decisions can still change. In practice a situation that is wrongfully detected can still change to a correct detection when the scene gets clearer.

After deciding between system 1 and system 2, selecting an action and checking whether this action is possible the motor system will send the command for that action to the robot. That doesn't mean the robot will always perform that action. The robot is inhibiting the execution of an action while it is already performing one of the following actions: turning left, turning right, turn around or crossing straight. These are all closed loop actions and cannot be interrupted. This was done to prevent unwanted side effects. The networks are not trained to recognise

objects or situations while performing these actions. Result from the networks are probably not reliable in such situations, especially in turning situations. This will lead to undesired actions and inhibition prevents that from happening.

6

Training

To reach the final goal of letting the robot walk a learned sequence, all the different networks have to be trained. This has to be done in stages and this chapter discusses the procedures needed for each stage together with the training results.

6.1 Stage 1 learning objects and system 2 activation

The two neural networks described in sections 5.2.1 and 5.2.2 are combined to a single neural network structure, because they use the same input information. This also means that a single dataset must be created for the training of the complete structure. To make the creation of training datasets possible, Simbad's basic functionality was extended with extra functionality for that purpose.

A ”Train” button was added to the control panel. Once this button is pressed a panel comes up with the different options that can be learned. The panel used in this first stage is divided into two parts. The objects/situations for decision making and the need for attention(System 2 activation). The different objects are presented in a more humanly understandable form, so for example the decision between left and right is called a t-junction.

Since the robot isn't able to operate on it's own in the test world at this time, manual control options were added to Simbad's functionality. This way the robot can walk to any location within the test world.

The procedure for creating training sets is as follows:

1. Walk to the object that needs to be added to the training set.

2. Position it in the right way.
3. Press the train button so the panel pops up.
4. Choose the object it needs to learn, whether the robot is in a higher attention(System 2) situation and press submit.
5. Repeat this procedure multiple times for the same object from different positions.

When a user presses the submit button data is gathered from the robots 24 sonar sensors. This data is then transformed together with the users input into a form that is suitable for Emergent and then sent to a dataset in Emergent. This way you are directly creating a training set from within the virtual world.

In practice all of the different object were trained from nine positions. In figure 6.1 can be seen how this was done for a front left situation. This figure shows that the front left checkbox is enabled and also that attention is checked.

The same is done for all the different types of objects in the test world. It must be said that every object was learned just once, thus not learning every left turn, but just a single one from different positions. By using this procedure the neural network should learn to recognise these objects/situations in a spatial invariant manner.

6.1.1 Result of the training

The goal for the training is to get to a level of zero errors three times in a row. Three times in a row was chosen to make sure the network was trained in the right way and that a zero error situation wasn't just a lucky shot. Training uses randomised data for each epoch and the error measurement was done by measuring the average error sum of squares, average SSE.

A zero error result should be possible when the network can make clear representations of the objects that it needs to learn. In practice it proved to be impossible to reach this level when the network was trained as a whole. Training had to be divided in first training the attention part and then the object recognition part.

To do this the S1,S2,S4,IT and What layer were lesioned, which means they are not active at the moment of training. Now training starts with only the input, S1 attention, S4 attention and attention layers enabled. It proved to be possible to let this part of the network train to a level of zero errors. At this moment network weights are saved, the lesioned layers are enabled again and the network is rebuild. Now the network already has the attention part trained and can start

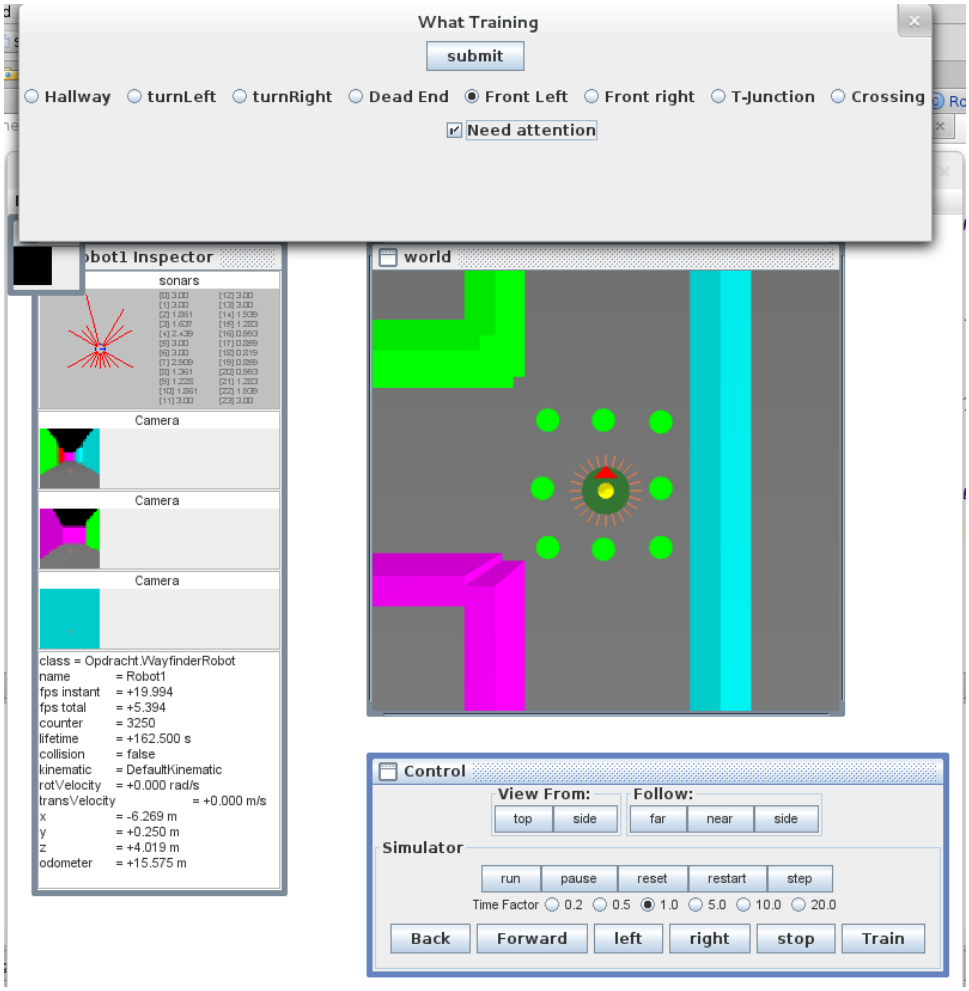

Figure 6.1. Create training set for object recognition. The green dots and the robots position show the places where the sonar information about that object is gathered.

the training for the rest of the network. Because of it's large number of neurons it took some time to train, but it also managed to get to a level of zero errors.

Figure 6.2 clearly shows the result of this two part training. It reached the three times zero errors in a row after 25 epochs. At this moment the rest of the layers were enabled and immediately the average SSE level shoot up to a level of around 1.4, but rapidly decreased after that. After that rapid decrease, fine tuning took some more time. As the figure shows it reached it's first zero error situation at around 60 epochs, but it took until 84 epochs to finally reach a situation of three times zero errors in a row.

6.2 Stage 2 training for action decision

Creating a training set for the network described in section 5.2.3 works in a similar fashion as described above. The robot is navigated through the world using the navigation buttons. This

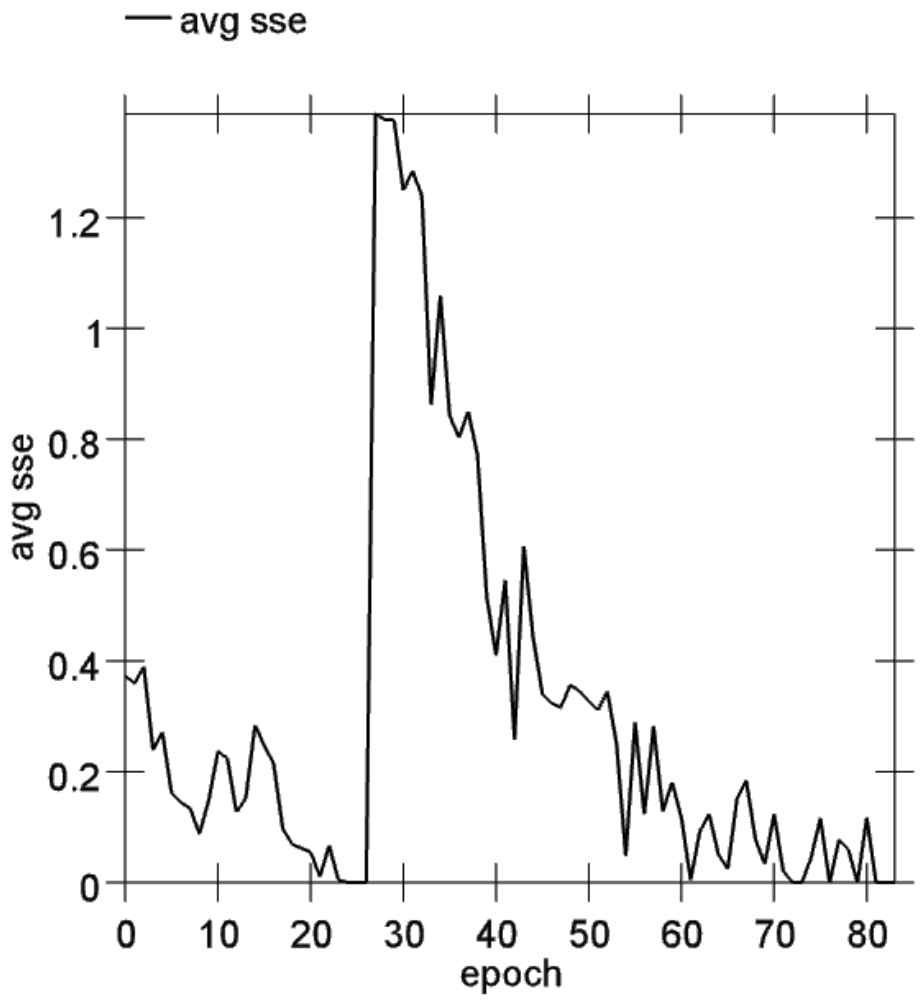

Figure 6.2. SSE first decreases to zero after the training with the object recognition part disabled. After enabling this part, the SSE shoots up but finally decreases to zero. With this procedure the combined object recognition and attention network can be trained.

time a different training method is attached to the train button. When pushed a panel becomes visible with just one option, a check-box with the label ”pass”. Enabling this option means it is the right moment to perform an action. Disabling means the opposite. Because the network needs to learn to discriminate between the right and the wrong moment to perform an action, both options need to be present in the dataset.

Figure 6.3 shows the procedure for learning the right and wrong moments. In this situation the right location and wrong locations for taking a left turn are added to the training set. The green dots show good locations and the red dots the wrong locations. The same procedure is used for all the different places in the world where the robot needs to perform an action other than moving forwards.

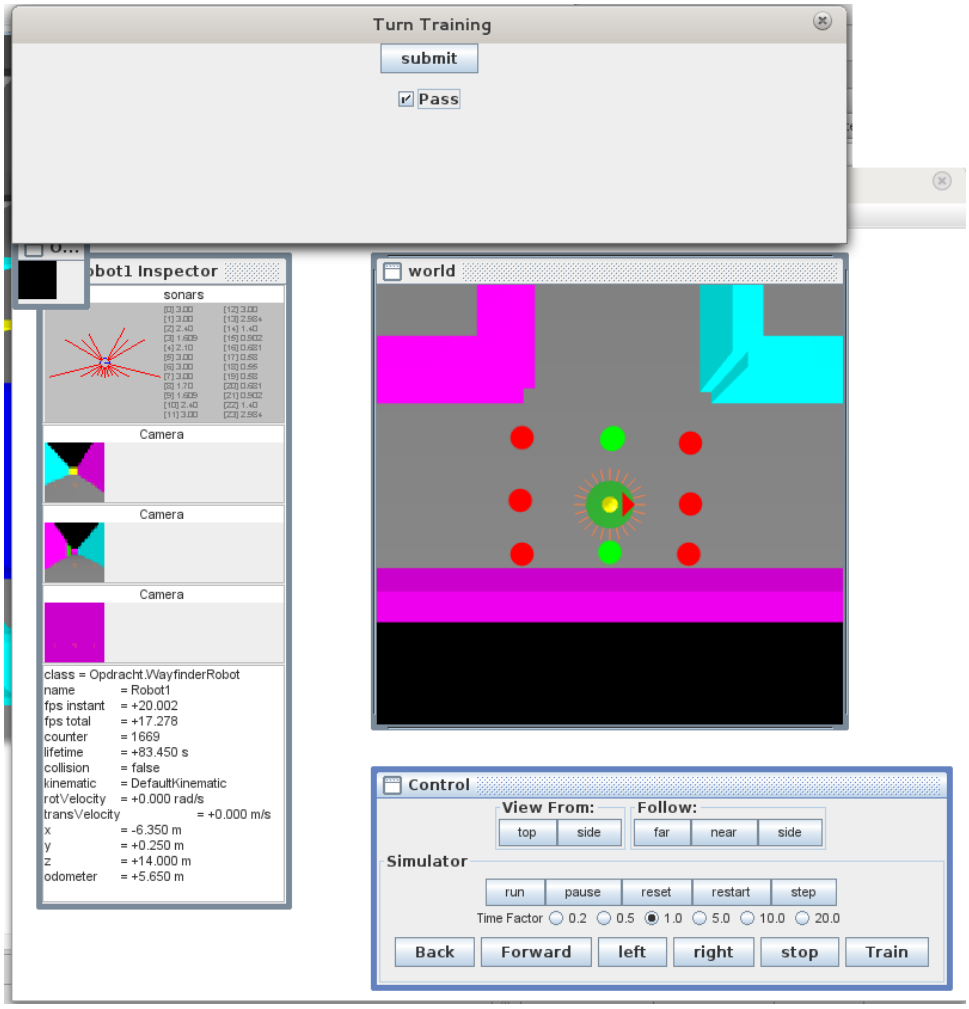

Figure 6.3. Procedure for training the right moment for action. Green dots show good moments, red dots wrong moments.

6.2.1 Result of the training

When the submit button is pushed, it gathers data from all 24 sonar sensors and combines this with the user input to create data that can be sent to an Emergent dataset. When the creation of the dataset is finished, the network will be trained using randomised data and should reach a zero error situation six times in a row.

The network proved to be quite capable of learning these situations. Figure 6.4 shows the result. It took just seven epochs to reach the error value of zero and stayed at zero from that point. So somewhat surprising training took just 12 epochs.

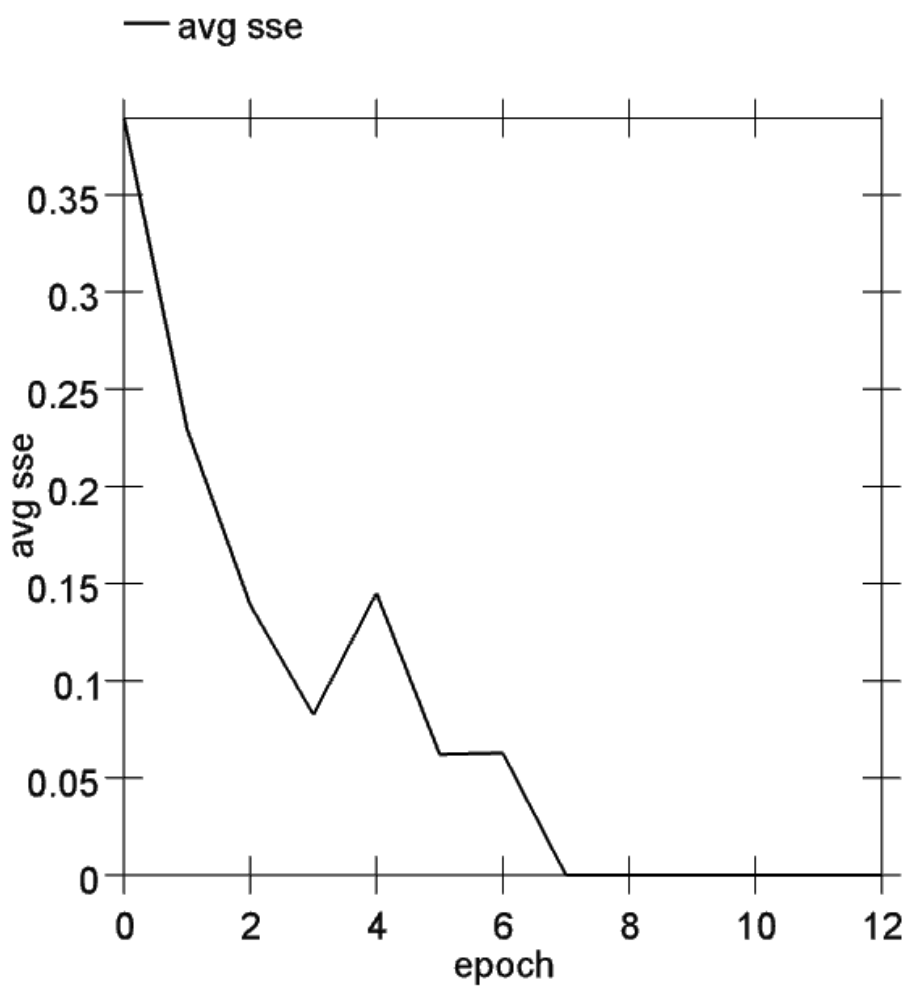

Figure 6.4. Rapid decrease of SSE to zero after just seven when the action decision network is trained.

6.3 Stage 3 testing the neural networks and the decision making model

At this moment almost all parts of the decision making model are built and trained. With the java implementation and the networks trained it should now be possible to detect the different decision situations, determine whether system 2 should be activated or not and when it is the right moment for action. The only thing missing here is a way for system 2 to take a decision. Before moving on to location detection and sequence learning what was build so far needs to be tested in practice.

To do this, two different implementations of system 2 were made. One is letting the robot choose a random decision and the other is asking a user to make the decision. The random implementation was built mainly to see how long the robot could walk around the world before

anything went wrong.

The other implementation of system 2 will ask the user which direction the robot should go when it has reached a location where it has two or more options to choose from. This provides a way to test the model and all the previously built and trained neural networks in practice. When the robot reaches a location, it should only ask for directions when system 2 is activated and it has reached a location where a green light was given to perform an action. When these criteria are met, a panel will pop up showing which directions are available at that moment. This way it is possible to see if it has detected the right situation.

Figure 6.5 shows an example of this for a t-junction. It was correctly detected as a system 2 type of situation and the robot stopped at a moment at which it can perform both available actions. The pop up shows that it has correctly identified this situation because it disabled the option to go straight and only gives the choice between left and right.

In essence what was built so far is almost a complete working model except for the system 2 decision making which is now left to the researcher. Before moving on to stage 4 it is important that everything is working as it is supposed to. Reaching this level was not that easy. Creating, training and testing the networks was an iterative process that needed quite a few cycles before reaching the reliability needed to proceed to the next stage.

6.4 Stage 4 training locations and sequence

At this moment in time the robot can correctly identify situations, activate system 2 when necessary and determine when to act. For sequence learning based on the recognition of locations it is necessary that it needs to learn for every situation with two or more decision options at which location it is currently at.

Since the robot has evolved quite a lot a this moment, it can now also help in creating the training dataset for the location network. a new system 2 implementation was build for this. The code that was used in stage 3 for testing was extended with training abilities. Now when the robot reaches a situation at which system 2 is active and performing action has been given green light a similar panel as in figure 6.5 shows up. It has the same direction buttons, but this time it also has an extra line of radio buttons with the labels A through M. These letters correspond with the letters in the virtual world. Now when choosing one of the possible directions the robot will move in that direction but will also collect data from the camera and sonar sensors. This information is then combined with the chosen location and sent to an Emergent dataset.

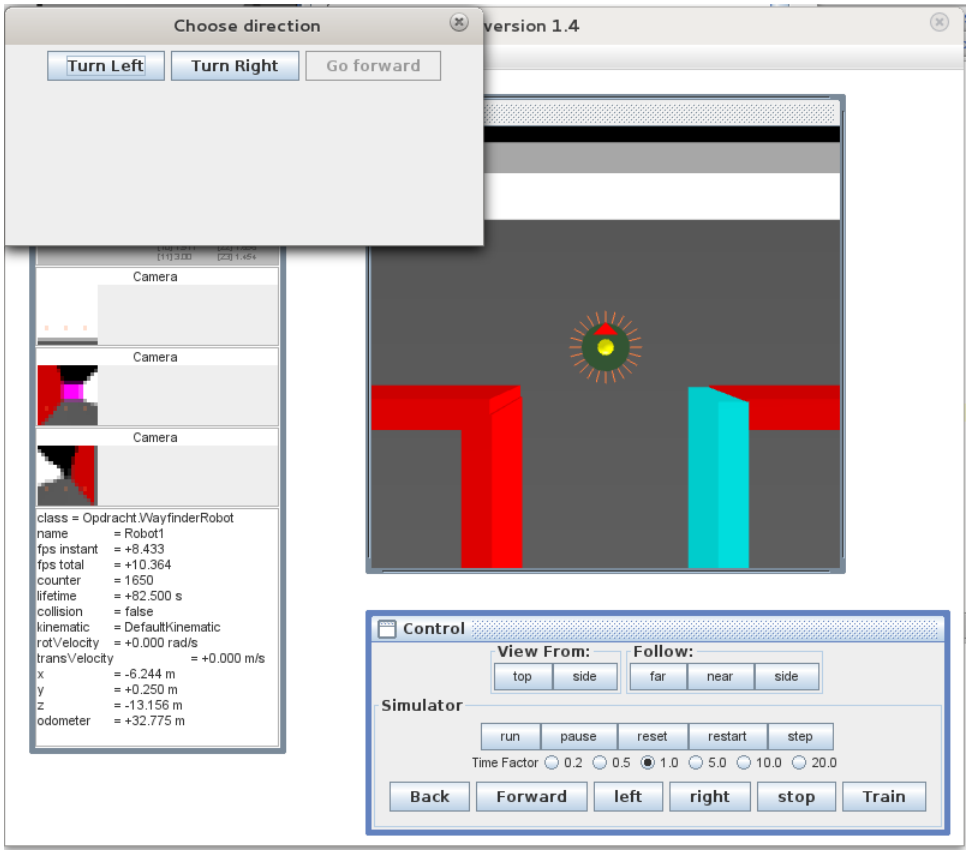

Figure 6.5. Deciding for the robot which direction to take. Unavailable options are greyed out

To learn sequences an extra option is added to the functionality described above and that is to remember the choices made. The chosen location is then remembered together with the direction that was chosen. At first the focus was on correctly learning locations so this option was disabled at first, but was turned on later when it was time to learn sequences. Figure 6.6 shows the panel for the robot asking for directions and the correct location. In this case location H.

6.4.1 Result of the training

At first the location network didn't manage to learn the different locations. It turned out that in the virtual world the locations were not unique enough to correctly discriminate between them. Sometimes it did manage to reach a zero error situation, but when tested in practice wrong decisions were made. At other times training couldn't even reach the zero error situation. The solution for this problem was quite simple, change some of the wall colors to make the locations more unique. This was a trial and error procedure and eventually led to the network being able to correctly identify all the different locations.

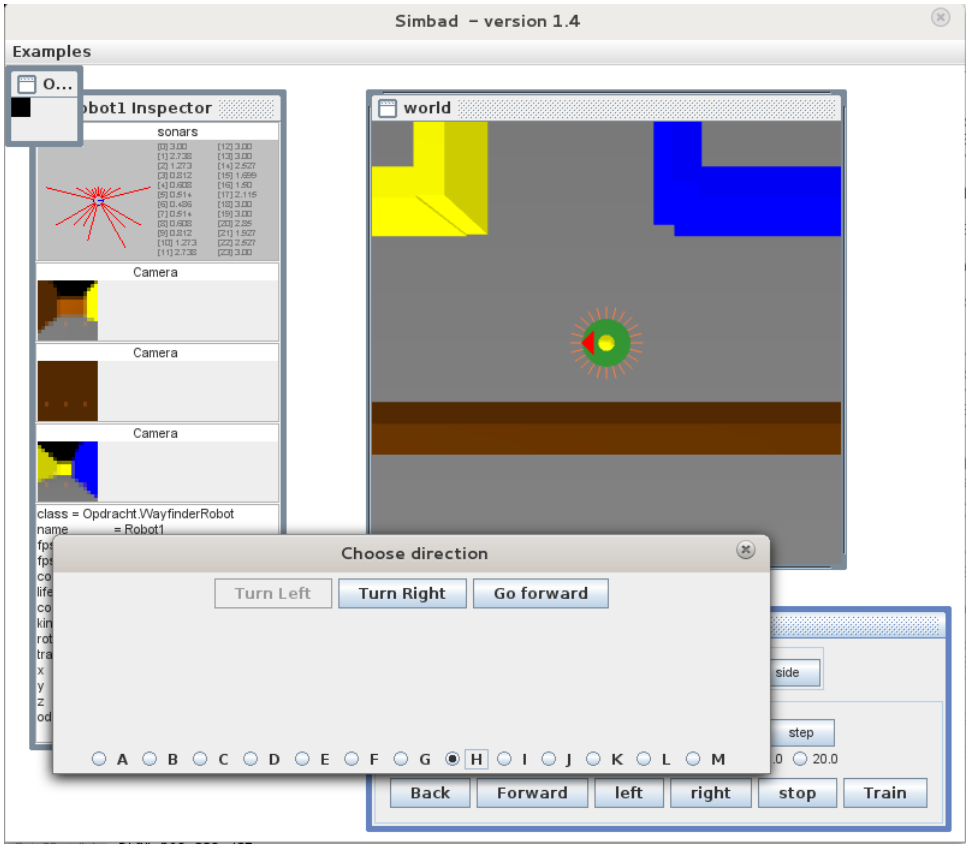

Figure 6.6. Creating training sets for location learning and optional for sequence learning. The chosen direction is then remembered together with the letter corresponding to the current location the robot has to learn.

Figure 6.7 shows the result from the network with the added sonar input. The dataset used for training is based on the route described in section 3.2. Data is randomised and trained until it reaches zero error three times in a row.

6.4.2 Sequence trainings

The final step is to learn a sequence of locations and decisions. The method of gathering data for training is the same as for learning locations. The difference is that now the chosen location and direction are combined into an input dataset that can be used to train the sequence network from section 5.2.6 with.

The route described in section 3.2 was walked and the location data was sent to the dataset for location training and the combinations of locations and directions were sent to a dataset used for sequence training. This led to 21 combinations of locations and directions that needed to be trained.

The procedure for training here is somewhat different than for the other networks. Since it is

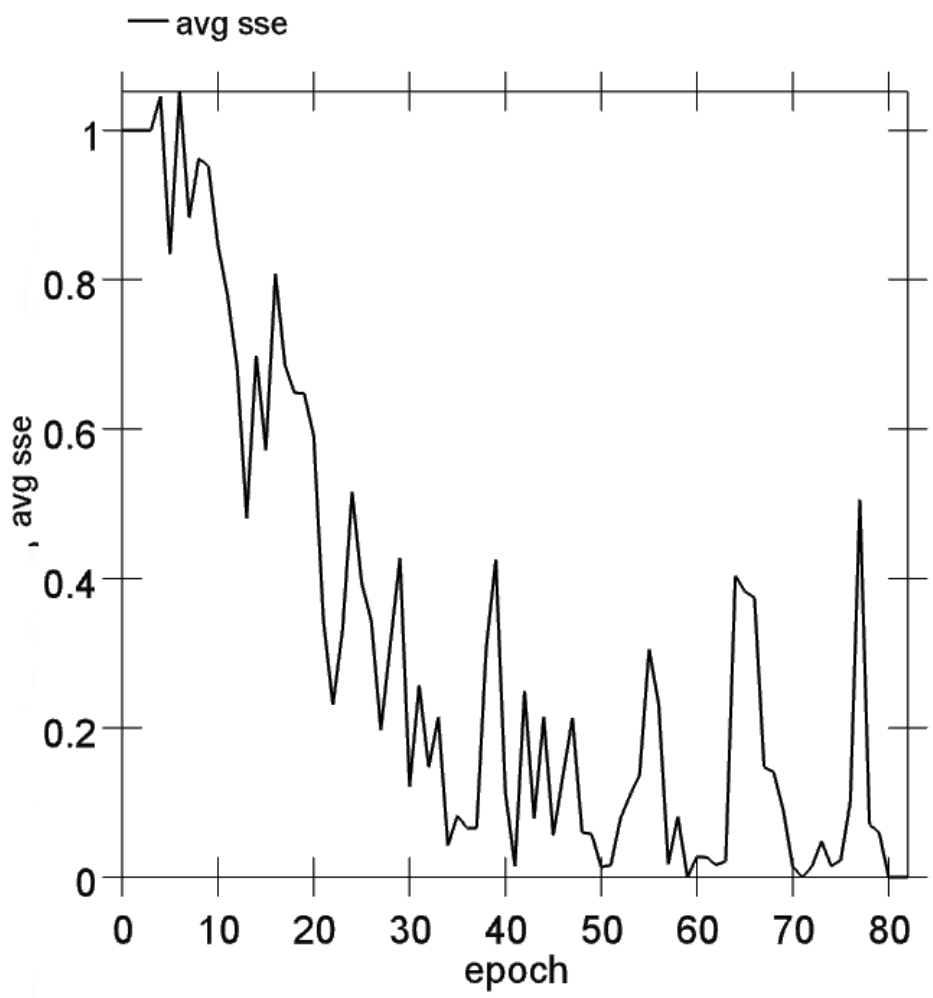

Figure 6.7. Results from training the location detection network with the extra sonar processing capabilities. These results show that learning locations was difficult, but possible after 80 epochs.

important to learn a sequence, data is not randomised before each epoch, but offered in the same sequence as the data was gathered. Training was finished when it reached a zero error situation three times in a row. Figure 6.8 shows clearly that the network was quite capable of learning this sequence. Zero errors was already reached after 4 epochs and stayed that way.

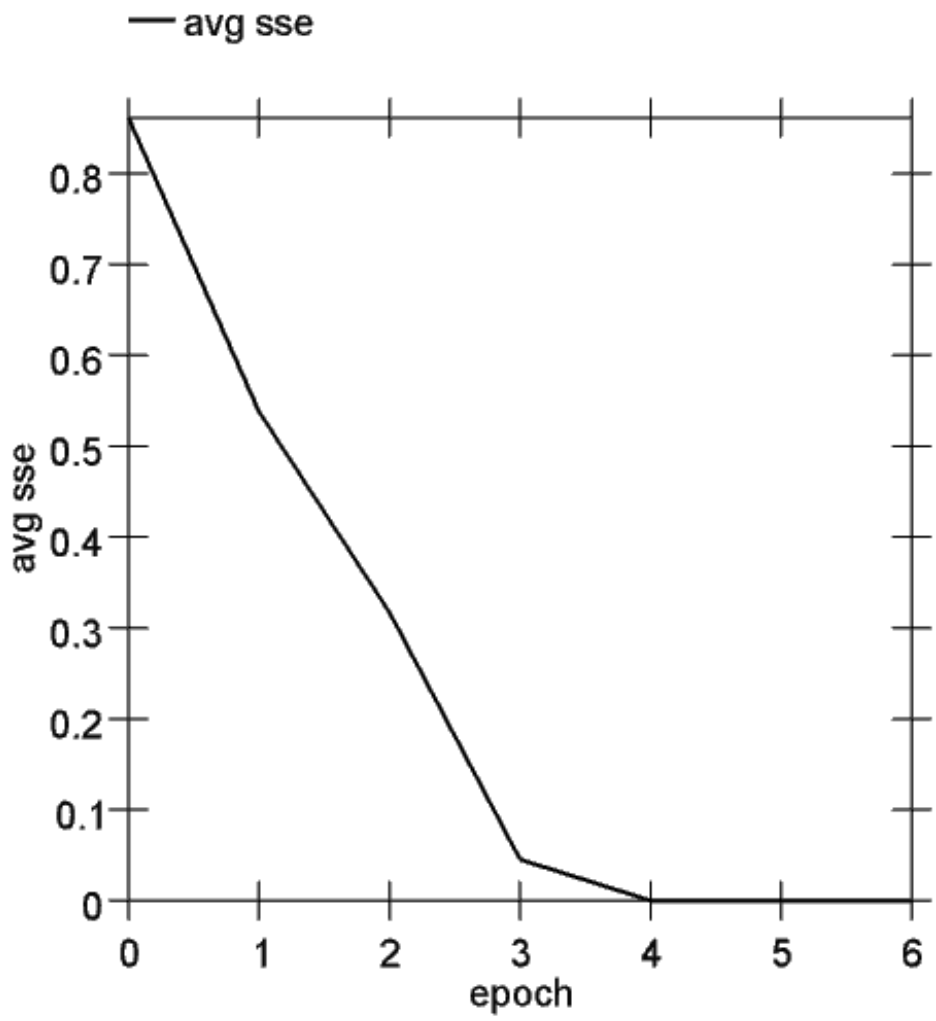

Figure 6.8. Results from the sequence training. The Elman network had no trouble learning the sequence.

7

Analysis

To test the robot in practice and analyse the different parts of the model the robot has been given the complex task of learning a route based on the recognition of locations/situations and the action associated with that location. All of the network structures were trained and a route was learned. This chapter focusses on analysing the results from the route learning task. In the first section of this chapter the different ways of data gathering are explained. In the following sections the actual analysis will be performed. This analysis will use a bottom-up approach by first analysing the different neural network structures, followed by the decision making and motor control and finally the results coming from the route learning task. Results will come from statistical analysis as well as lesioning and observation. All the statistical analysis will be done using R (R Core Team, 2013).

7.1 Producing output

To make analysis possible, output from the different parts of the model is necessary. This output is produced at different levels. At the highest level the robot's route will be tracked and output from the decision making model will be recorded. At the lowest level neuronal activation from different layers of the neural networks will be recorded.

7.1.1 Output produced by the robot

To help determine how well the robot learned the route, extra functionality was programmed to keep track of the robot's location and to follow the decision making process. Figure 7.1

shows the output screen that was created for this purpose. It consists out of a map which tracks the robot's position and a text area which records the decision making process. The map is constantly updated with the robot's location. A green line shows the route the robot has walked. Next to the line, red numbers indicate places where the robot has performed some action based on decision making. These numbers are automatically created and placed in the map every time a decision from either system 1 or system 2 was turned into an action. These numbers automatically increase after each of these actions.

Output from the robot's decision making process was sent to the text area next to the map. It gives output from the different structures in the model. First it shows what both system 1 and system 2 have decided, second the decision made by the motor system, third the output telling the robot whether the decision from the motor system has the permission to be performed or not and fourth whether the action that has been given permission can be performed or that a previous action inhibits it from being performed.

When system 2 is set up for walking a sequence the output frame also shows the location that it recognized and what the action is that the robot wants to perform at that location. The numbers shown in the text area correspond to the numbers placed on the map. To give an example of that, take the route walked in figure 7.1. when going right at location C, which was assigned the number 10 everything that has the number 11 belongs to the decision making process for the next action to perform. In this case the robot is walking to location D and the decision made at number 11 is taking a left turn at the t-junction. After that the number increases to 12.

The robot also produces another form of output. It creates a datafile with frame-count, system 2 activation and location info. Frame count is time information, system 2 activation has a value of 0 for no activation and a 1 for activation. Location is the same location number as described above. This was done so a time plot can be made that shows system 2 activation over time and locations were added to determine at which locations system 2 was activated.

7.1.2 Output produced by the neural networks

To determine whether and how representations are formed within the individual layers of the sonar and location detection networks described in sections 5.2.1 and 5.2.4 principal components analysis (PCA) will be performed on both networks. To create output for this, the Emergent programs that control the networks are extended with monitors that record the

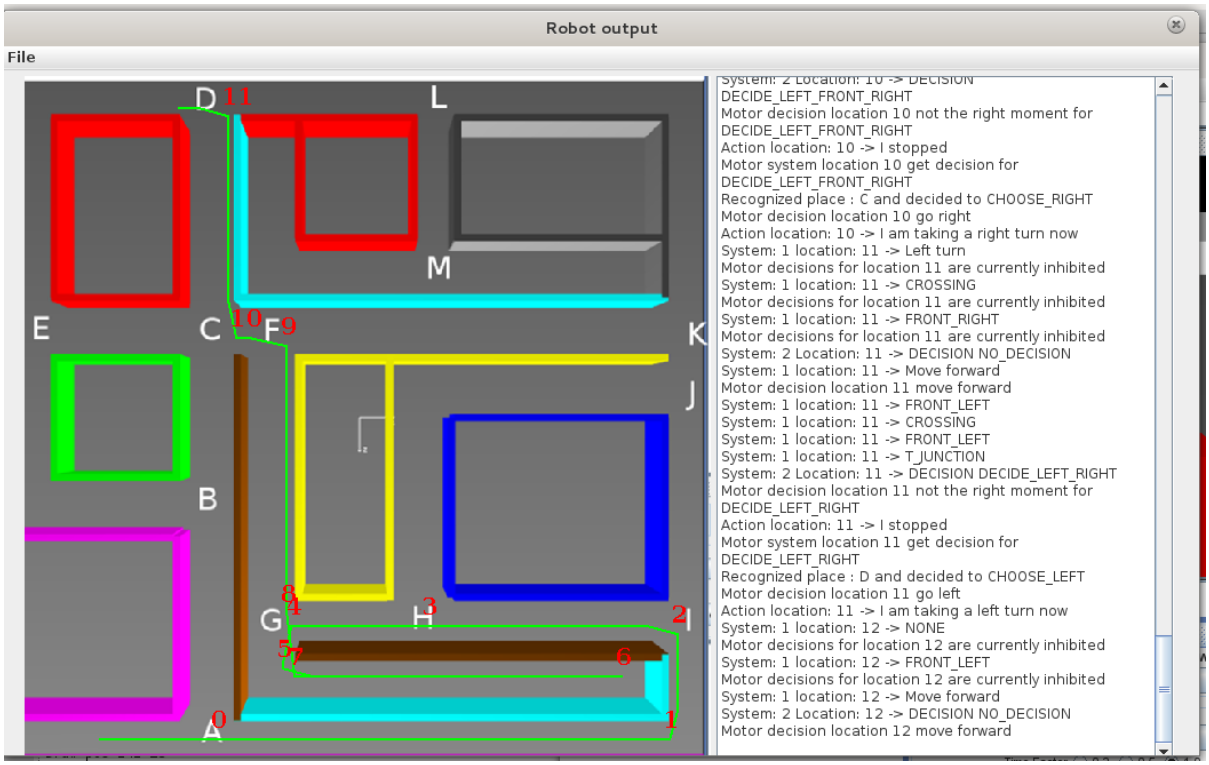

Figure 7.1. Robot's output screen. Tracking the robot's location and decision making. The green line shows the route the robot walked. The route started at the lower left corner. The red numbers indicate places where the robot performed an action after a decision from either system 1 or system 2.

activation from all the neurons within a given layer and save it to a dataset. These datasets can then be exported and imported into R for further analysis.

For the sonar situation detection network neuronal activation from the output, IT and S4 layers are recorded. For the location detection network neural activation from the output, IT and V4 as well as the S4 layer are recorded.

7.2 Analysis of the sonar detection network

The analysis was started by performing a PCA on the IT layer. The expectation was that this layer would have formed eight major components which are the direct representations of the eight output objects. Table 7.1 shows the importance of the first 10 components found and the scree plot in figure 7.2 displays these components with their eigenvalue. These results indicate that there are only three components which have a standard deviation and eigenvalue larger than one. These three can be considered as the most important components and that number is much lower than the expected eight.

The next step is to correlate these components to the activation of the output neurons. Table

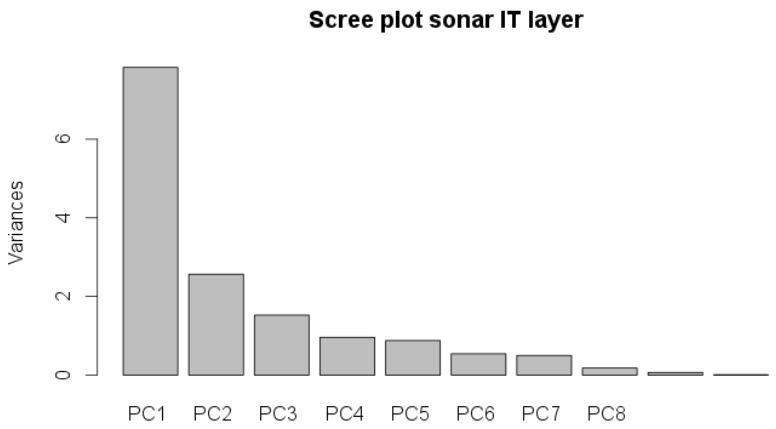

Figure 7.2. Scree plot for the IT layer, shows the importance of the different components found by the PCA analysis.

Table 7.1
Importance of the First 10 IT PCA Components

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Standard deviation	2.80	1.60	1.24	0.98	0.94	0.74	0.71	0.43	0.26	0.11
Proportion of Variance	0.52	0.17	0.10	0.06	0.06	0.04	0.03	0.01	0.00	0.00
Cumulative Proportion	0.52	0.69	0.79	0.85	0.91	0.95	0.98	0.99	1.00	1.00

7.2 shows the correlations which have an absolute value larger than .5. This gives a somewhat more clear picture. It definitely shows that there is a single component (PC1) which is the representation of a hallway. With a correlation of -.97 it is safe to say that. The same thing can be said about a dead end, which has a correlation of -.90 with PC8.

For the other situations things are not that simple. When you only look at the correlations between the IT layer components and the activation of the output neurons it does seem that there are 8 important components in the IT layer instead of the three found in the PCA on the IT layer. The correlations between the IT layer components and the activation of the output neurons is not in general high enough to state that these components are the direct representations of the learned situations. It might be that these components represent more abstract forms that are shared by different objects. That could explain why PC1 has a correlation of -.97 with a hallway and an opposite correlation of .64 with the front left situation. So a part of the information learned about the hallway can be used to recognise the front left situation. Something similar might be present with PC2 which has a negative correlation of -.66 with the front left and a positive correlation of .78 with the front right situation. This can be explained as a certain

Table 7.2
Correlations Between Sonar IT Components and Output Neurons in the Sonar Detection Network

	Hallway	Turn left	Turn right	Dead end	Front left	Front right	T-junction	Crossing
PC1	-0.97				0.64			
PC2					-0.66	0.78		
PC3							0.72	
PC4							0.62	
PC5			0.51					-0.80
PC6		0.74						
PC7			0.72					
PC8				-0.90				

feature being present or not. For instance having a wall on the left might be such a feature.

Why then does the PCA show a number of three components while the correlations between the two layers indicates that there are eight important components? The answer is quite simple. These results are based on the route the robot walked. In these results some situations are more present than others. Most of the time the robot walks through hallways and this explains why the first component explains most of the variance. The front right situation is then visited most and this has the highest correlation with the second component. So the first PCA doesn't show how much representations were formed in the IT layer and how important they are, but rather their frequency of presence in the data. The correlations between these components and the output activity is then needed to show that there are actually eight components formed which can be seen as representations. Only for the hallway and the dead end it can be said that these components probably are the direct representation of these situations. For the other situations this picture is not that clear and some more abstract representations might have formed.

In the next fase of this analysis we will go one layer deeper and perform a PCA on the S4 layer. The scree plot for this analysis is shown in figure 7.3. This figure shows similar results as the IT layer PCA. The high importance of the first component indicates that this probably is the hallway also. Correlations are then calculated between the S4 components and activation of the output neurons as well as with the IT layer components. This should give an idea about whether some representation have already formed in the S4 layer or that maybe some of the IT components get divided into multiple S4 components. This division into more components in the S4 is something you might expect from a ventral stream based structure where the RF increases with the passing of each layer and representations become more and more abstract. Table 7.3 shows the correlations between the S4 layer components and the output layer neurons

and table 7.4 shows the correlations between the S4 and IT layer components.

What is clear from these results is that the first component found has a correlation of .97 with the first output neuron. Apart from the sign, this is exactly the same result as the correlation between the first IT factor and the output. When the S4 components are correlated against the IT components, an interesting picture emerges. What stands out is the fact that the first component in both layers correlates for 100%, indicating that this is exactly the same factor. This proves that the representation for the first output neuron, the hallway, is already established in the S4 layer and remains exactly the same while passing through the IT layer.

Somewhat the same situation is present for the representation of the third output neuron which codes for a dead-end. PC10 from the S4 has a .80 correlation with this neuron. PC10 from the S4 layer also has a correlation of .95 with PC8 from the IT layer and PC8 from the IT layer has a .90 correlation with this neuron. With even less certainty this might even be the case for detecting the front right situation which is represented by neuron five. It looks like some of the representations, probably the ones that are easier to learn, are already present in the S4 layer and the extra processing from the IT layer doesn't add anything to improve on that.

The other situations are probably more difficult to distinguish and need a combination of IT and S4 representations to activate the right output neuron. For example PC5 from the IT layer has a high correlation of -.80 with output neuron seven which codes for a crossing. When the components from the S4 layer are correlated with the activation of this output neuron no real strong correlation between one of the components and the output activation emerges. Also no strong correlation exists between the PC5 component from the IT layer and one of the S4 components. This means that for a crossing the representation is formed in the IT layer and isn't already present in the S4 layer. The same thing looks to be true for the representation of a t-junction and a left turn. These two also have quite strong correlations between IT components and output neuron activation, but fall apart into separate components when compared to the S4 layer.

For the front-left situation and right turn there is not one dominant IT component but two components which have somewhat lower correlations. In absolute numbers between .5 and .6. When these two components are correlated to the S4 components, the IT components have at most absolute correlations of around .6. In these situations no clear representation exists in the IT or the S4 layer. These situations are correctly recognized when the robot is tested in practice, so clearly the combined output of the S4 and IT layers produce the right output. This means that representations of these situations is made up out of the sum of the components present in the

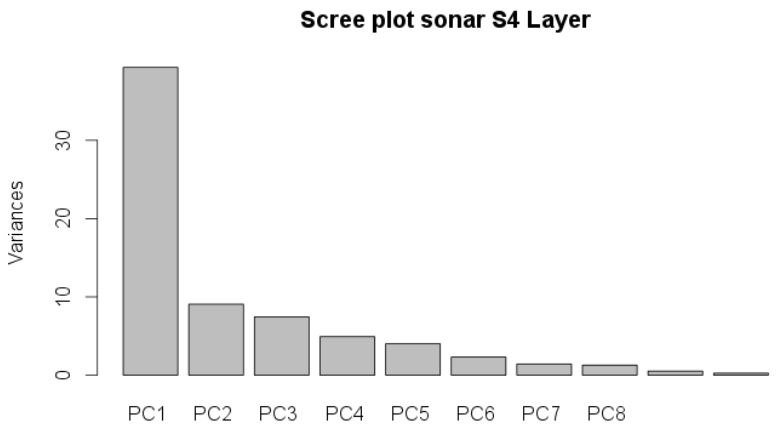

Figure 7.3. Scree plot for the IT layer

IT layer and a single representation can only be present in the output layer.

Table 7.3
Correlations Between Sonar S4 Components and Output Neurons in the Sonar Detection Network

	Hallway	Turn left	Turn right	Dead end	Front left	Front right	T-junction	Crossing
PC1	0.97				-0.63			
PC2					0.53	-0.82		
PC3								
PC4							0.84	
PC5			-0.63					
PC6		-0.63						
PC7			0.51					0.55
PC8								
PC9								
PC10		-0.50		0.81				

7.3 Analysis of the location network

The location detection network comes in two variants, one with and one without sonar input. Two hypothesis will be tested here. The first is that color information alone is sufficient to correctly learn all the locations and the second is that shape information provided by the sonar will improve location detection results. To perform this analysis the robot reports it's detected location in the same way as it would do when controlled by the sequence implementation. The only difference is that for analysis the robot will not ask the sequence learning network for the

Table 7.4
Correlations between S4 and IT Components in the Sonar Detection Network

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
PC1	-1.00							
PC2		-0.93						
PC3			0.72					
PC4			0.56	0.60				
PC5				0.61	-0.66			
PC6					0.51	-0.63		
PC7							0.79	
PC8								
PC9								
PC10								-0.95

next direction, but the researcher. This way the researcher can let the robot walk the same route even while it can make wrong decisions and it's finding can be reported for each location.

When trained with the same data set, both networks managed to get to the required level of three times zero error in a row. At this moment it looks like there is no advantage of adding the sonar information and just visual information should be sufficient. However when put to the test in practice there is a difference. Table 7.5 shows the results when they are tested in the virtual world. They perform almost the same except for location D which is stably detected by the network with sonar and less stable by the network without sonar. This effect is even stronger for location A when that is visited as last location in the sequence. Even more problematic is the fact that it even detected one location wrong, it identified location I as J. Since the decisions made by this network directly influence the decision making, it was decided that the network extended with sonar will be used for learning the sequence.

7.3.1 Lesions

To test what the effect is of adding extra sonar layers to the location detection network, several lesion studies were performed. When the network uses color information to form representations for each location, then disabling one of the three color input layer would effectively mean that it would become color blind for that specific color. This is tested by lesioning the red, green and blue input layer and recording which locations it then detects along the route. To see how much effect adding sonar information has, the sonar layer shall also be lesioned. Finally the sequence is walked with all the color input disabled and only the sonar will be active. The robot is manually controlled in the same way as described above and walks the same route every time.

Table 7.5
Comparision Location Detection With and Without sonar

With Sonar	Without Sonar
A	A
I	I
H	H
G	G
G	G
F	F
C	C
D	L/D
E	E
B	B
C	C
D	D
L	L
M	M
M	M
L	L
K	K
J	J
H	H
I	J
A	J/G/A/L

Table 7.6 show the results for the different lesions. The first column shows the results without lesions. The next three columns the different color input lesions, followed by lesioning the sonar and last lesioning all color.

The results from the lesioning of the color input clearly shows that the network has used color for learning. Disabling red led to 9 errors, disabling green 10 and disabling blue 13 errors. It is no surprise that when red is disabled that it had no problems detecting locations K and J and has more trouble with locations A and B. What is surprising is the fact that it did correctly recognise locations D and B. Maybe red is not the dominant decisive factor here.

A thing that was expected is that the result from lesioning green and blue have more in common than comparing red and blue or red and green. Because green is a combination of blue and yellow there should be more overlap between them than with red. When blue is compared to green they have 12 detected locations in common. When red is compared to green this number is seven and compared to blue there are eight in common.

All these results clearly indicate that color information is used and is very import in recognizing locations. The next question is how much does the sonar input add to the results?

Table 7.6
Effects of Lesions on the Different Input Layers

No Lesion	Red	Green	Blue	Sonar	Colour
A	B	A	J	E	M
I	I	I	I	I	A
H	J	L	L	A	L
G	H	G	G	A	I
G	K	L	L	A	K
F	F	K	F	F	A
C	C	K	K	A	C
D	D	D	D/L	A	H
E	E	A	C	H	M
B	H	I	G	A	I
C	C	C	C	A	K
D	D	D	L	A	I
L	H	L	L	A	A
M	M	L	F	A	L/I
M	M	M	M	A	M
L	D	L	L	A	L/I
K	K	G/K	I	A	A/K/B
J	J	J	I	A	K
H	H	I	I	A	I
I	I	I	I	I	I
A	J	L	L	A	A

If this influence is only marginal then it is expected that lesioning this layer doesn't do much damage to the results. However the opposite is true. Performance becomes much worse than disabling one of the color input layers. Just four locations were detected right which clearly indicates that the network is seriously handicapped without the sonar layer. It almost performed as worse as the last experiment where all color information was disabled. This led to only three locations that were detected right. By no doubt representations are formed based on a combination of visual and sonar input and they are both equally important in the forming of those representations.

7.3.2 Principal components analysis

The results from the lesion experiments should be backed up by the results from the principal components analysis (PCA).

First a PCA is performed on the activation of the IT layer neurons. Figure 7.4 shows the scree plot and table 7.7 shows the importance of the first 12 components.

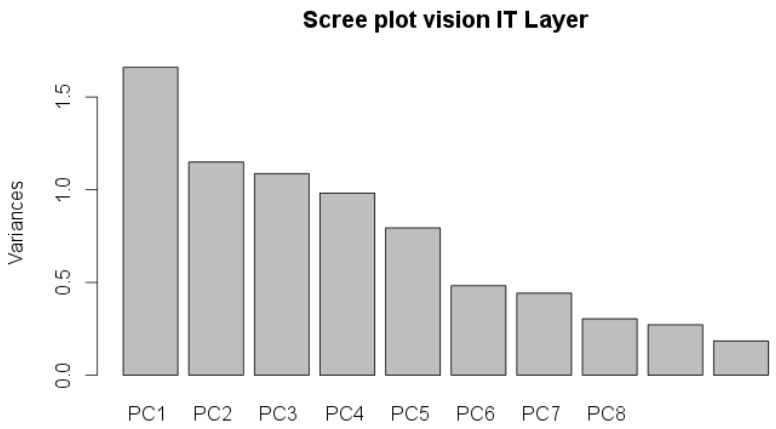

Figure 7.4. Scree plot for the IT layer

Table 7.7
PCA Importance First 12 components IT Layer

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12
Standard deviation	1.29	1.07	1.04	0.99	0.89	0.69	0.66	0.55	0.52	0.43	0.37	0.23
Proportion of Variance	0.22	0.15	0.14	0.13	0.11	0.06	0.06	0.04	0.04	0.02	0.02	0.01
Cumulative Proportion	0.22	0.37	0.52	0.65	0.75	0.81	0.87	0.91	0.95	0.97	0.99	1.00

One might expect that the components found in the IT layer would correlate highly with the activation of the output neurons. Such high correlations would indicate that a location is identified by just one or maybe a few other strong components. Table 7.8 doesn't show such a thing. Only correlations with an absolute value larger than .5 are shown. Correlations are quite low, at least lower than one would expect when strong representations were formed in the IT layer.

Maybe this has something to with the fact that the IT layer combines both the sonar and visual information? To dig deeper into this question PCA's are also performed on the S4 and V4 layers. The scree plot of the S4 layer can be seen in figure 7.5 and the scree plot for the V4 layer can be seen in figure 7.6. The components from the S4 PCA are correlated with the activation of the output neurons and the same thing is done with the components from the V4 layer. Table 7.9 show the results for the correlations between the S4 components and the activity of the output neurons and table 7.10 displays the correlations between the V4 components and the activity of the output neurons. The thing that stands out here is that they provide a very similar picture as the correlation shown between the IT layer components and the activation of

Table 7.8
Correlations Between PCA components IT Layer and Activation Output Neurons of the Location Detection Network

	A	B	C	D	E	F	G	H	I	J	K	L	M
PC1												0.55	-0.58
PC2			0.57	-0.58									
PC3												0.54	
PC4	0.75												-0.61
PC5							-0.62						
PC6				-0.55		0.53							
PC7						0.67							
PC8		-0.77											
PC9					0.53								
PC10								0.59					
PC11										0.60			
PC12											0.85		
PC21													-0.50

Table 7.9
Correlation Between S4 components and Activation Output Neurons of the Location Detection Network

	A	B	C	D	E	F	G	H	I	J	K	L	M
PC1													-0.56
PC2			0.57										
PC3												0.64	
PC4	0.66												-0.60
PC5									-0.51				
PC6				-0.62		0.64							
PC7							-0.73						
PC8		-0.56											
PC9			-0.56							0.73			
PC10		0.65			-0.58								
PC11								-0.65					
PC12											0.85		
PC21												-0.62	

the output neurons. Although there are some correlations higher than .80, most of them are quite low. Too low to conclude that strong representation were formed in the S4 or V4 layers.

The last analysis to perform is to see how well the V4 layer components correlate to the IT layer components and how well the S4 layer components correlate to the IT layer components. Maybe the interplay between the vision and sonar networks explains why IT component and locations have such low correlations.

Table 7.12 shows the correlations between the V4 components and the IT components. Table 7.11 shows the same thing but then for the correlations between the S4 and IT layer components. Only absolute values above .5 are shown here. Both tables show remarkably high correlations between the components. Another thing that stands out is the diagonal pattern that emerges in both tables and most striking in the correlations between the S4 and IT layer components.

The correlations between the S4 and IT layer components also display some extremely high

Table 7.10
Correlation Between V4 components and Activation Output Neurons of the Location Detection Network

	A	B	C	D	E	F	G	H	I	J	K	L	M
PC1													0.73
PC2									0.51			-0.52	
PC3	-0.80												
PC4			0.55										
PC5				-0.81									
PC6						0.82							-0.57
PC7							0.69						
PC8		0.58											
PC9		-0.60			0.54								
PC10										0.64			
PC11								0.56			0.57		
PC12								0.52			-0.53		
PC21									-0.56				

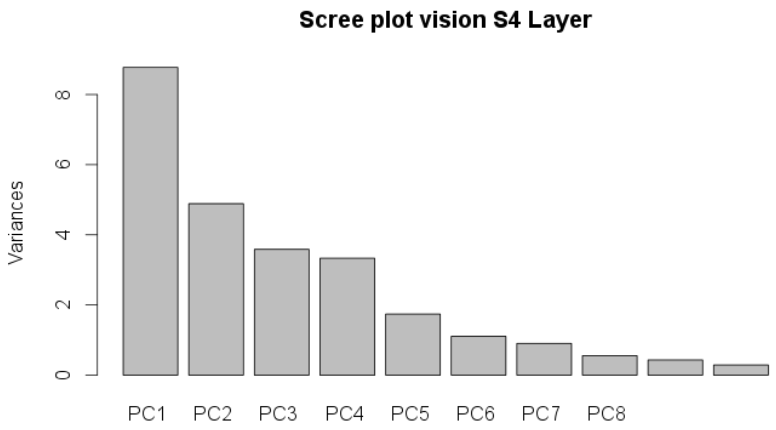

Figure 7.5. Scree plot for the S4 layer

values, 15 of them are, when taken absolute, larger than .8. To a somewhat lesser extends this is also the case for the correlations between the V4 and IT layer components. There are 10 absolute values larger than .8, indicating strong influences on the IT layer.

Then why then are all the previously calculated correlations so low? When looking at these results the answer seems to be that although both the Visual and Sonar part exert high influence on the IT layer, their influence is not always in the same direction. For example the correlation between the first component in the IT layer correlates with a number of .94 with the first component of the S4 layer. This is a very high correlation. The same IT component also correlates reasonably high with the first V4 component with a value of -.64. As you can see the S4 has a positive correlation and the V4 a negative one with this component. The same thing happens at more places and that might explain the low correlations.

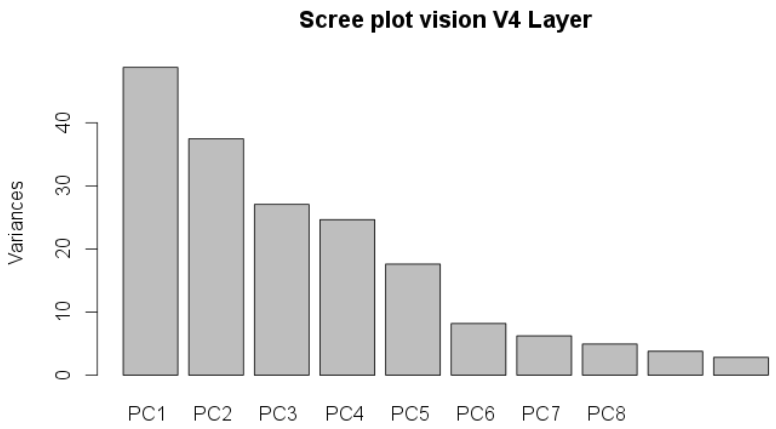

Figure 7.6. Scree plot for the V4 layer

The number of data is not enough to have any certainty about this, but it indicates that the representations of the different locations are the combined activations of the V4, S4 and IT layers. Not just a single representation for each location in the IT layer. This can also explain why the effect on the recognition of locations is so dramatically impaired when the sonar input layer is lesioned.

7.4 Sequence results

This last analysis will focus on how well the robot managed to learn the desired route as described in section 3.2. Figure 7.7 shows the image from the output window that tracked the robots route. The overall results are summarized in table 7.13. When looking at this table it is clear that the robot not only managed to walk the learned route but got a perfect score on the recognition of situations and locations.

This is of course the perfect result that was hoped for and is the end result of the trial-and-error process described in the previous chapters. Does this then mean that the robot is perfect in it's decision making? That's not the case. When you look at a part of the decision making as produced in the output window, it becomes clear that reaching a correct decision is not that easy. This part is shown in figure 7.8 and shows the decision making process leading to the decision for location F.

The correct situational object is a t-junction with the options to go left or right. As you can see it looks like the robot can't really make up it's mind at first. It starts with front left, then crossing, front left again, right turn and finally when it comes closer the picture becomes clearer

Table 7.11
Correlation between S4 components and IT components

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16	PC17	PC18	PC19	PC20	PC21
PC1	0.94																				
PC2		0.91																			0.51
PC3			0.88																		
PC4				0.88																	
PC5					-0.78																
PC6						0.85															
PC7							0.82														
PC8								0.86													
PC9									0.73	-0.60											
PC10											-0.84										
PC11										0.65											
PC12												0.96									
PC13													0.99								
PC14														-0.81	-0.54						
PC15																	-0.56	0.78			
PC16																	0.80	0.53			
PC17															0.64						
PC18																			-0.87		
PC19																-0.89					
PC20																				0.97	
PC21			-0.71																		

Table 7.12
Correlation between IT components and V4 components.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16	PC17	PC18	PC19	PC20	PC21
PC1	-0.68	-0.54																			
PC2				0.85																	
PC3		-0.50	-0.53		0.63																0.65
PC4	-0.64																				
PC5					0.53																
PC6						0.59	0.58														
PC7						0.59	-0.73														
PC8								-0.84													
PC9									0.89												
PC10											0.82										
PC11										0.76		0.61									
PC12												-0.68									
PC13															0.86						
PC14																-0.88					
PC15																			0.93		
PC16																	0.70				
PC17													0.85								
PC18																		-0.79			
PC19														0.91							
PC20																				-0.99	

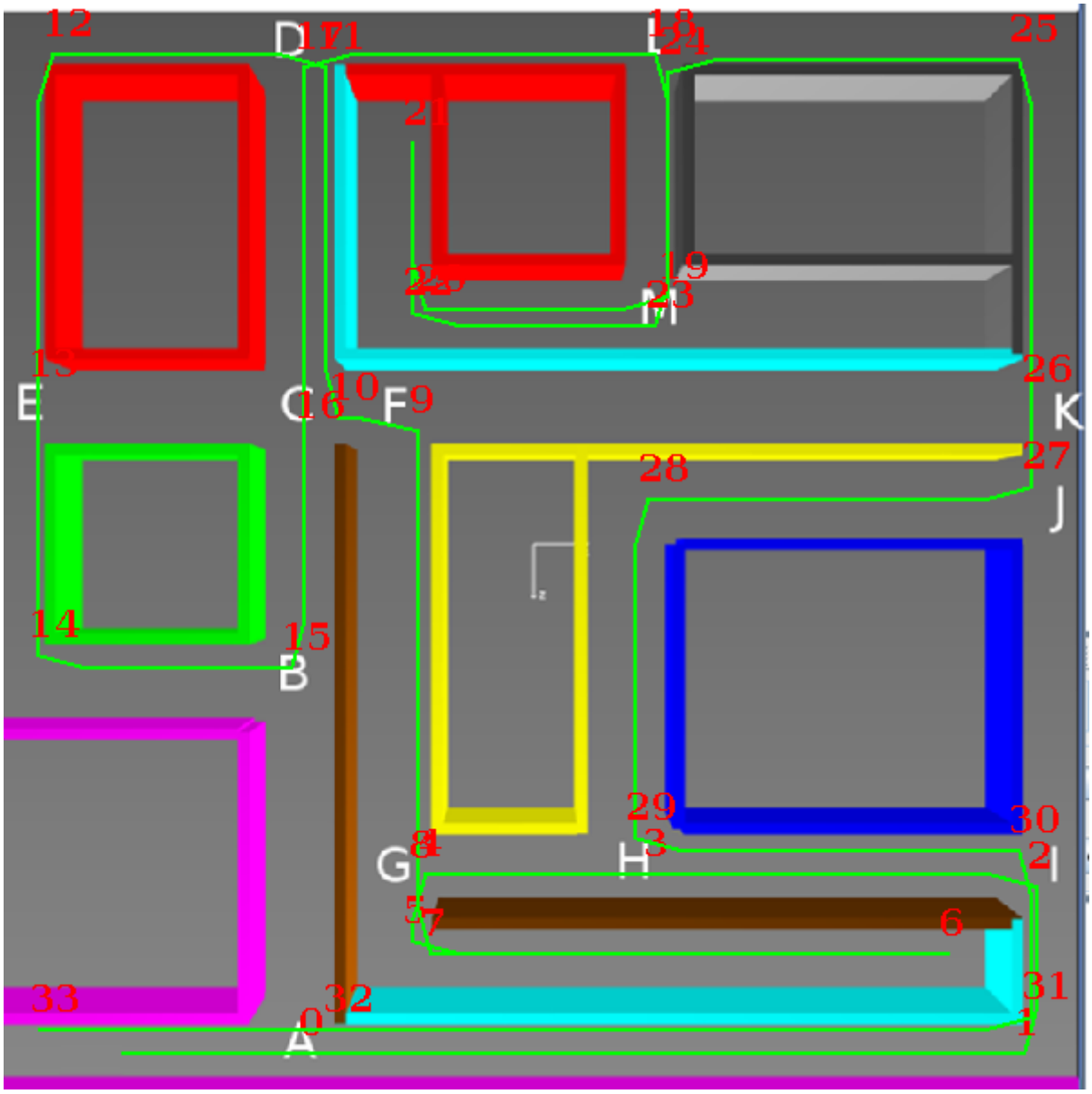

Figure 7.7. Output from the learned route. Green line shows the walked route and the red numbers indicate all the places where the robot performed an action based on either a system 1 or system 2 decision.

and the decision becomes the right one.

What can also be seen from this example is that when system 1 detects front right, front left or a crossing this is completely ignored since no automatic actions are connected to those situations. Only when system 2 comes in action, at line 9, it tries to perform an action based on the DECIDE LEFT RIGHT decision. That action would in this case be trying to detect a location, run the sequence network and finally moving into the correct direction.

Another thing that shows up at line 13, is that the motor system has decided that a decision must be made between left and right, but that it's not the right moment to do that. This is the

Table 7.13
Learned Sequence Results. Type L=Left, F=Forward, R=Right

Number	Location	Type of Location	Decision
0	A	FL	F
1		L	L
2	I	FL	L
3	H	FR	F
4	G	LR	L
5		L	L
6		D	T
7		R	R
8	G	FR	F
9	F	LR	L
10	C	LFR	R
11	D	LR	L
12		L	L
13	E	FL	F
14		L	L
15	B	LR	L
16	C	LFR	F
17	D	LR	R
18	L	FR	R
19	M	LR	R
20		R	R
21		D	T
22		L	L
23	M	FL	L
24	L	RL	R
25		R	R
26	K	FR	F
27	L	FR	R
28		L	L
29	H	LR	L
30	I	LR	R
31		R	R
32	A	FR	F
33		D	T

result of the action detection network preventing an action to be performed. Finally performing an action gets green light and the location detection can be started and that location can be passed on to the sequence network. In this case location F gets correctly detected together with the decision to go left.

```
System: 1 situation: 9 -> Move forward
Motor decisions for location 9 are currently inhibited
System: 2 situation: 9 -> DECISION NO_DECISION
Motor decision location 9 move forward
System: 1 situation: 9 -> FRONT_LEFT
System: 1 situation: 9 -> CROSSING
System: 1 situation: 9 -> FRONT_LEFT
System: 1 situation: 9 -> Right turn
System 2: situation: 9 -> Attention needed to detect an object
System: 2 situation: 9 -> DECISION WAIT
System: 1 situation: 9 -> T_JUNCTION
System: 2 situation: 9 -> DECISION DECIDE_LEFT_RIGHT
Motor decision situation 9 not the right moment for DECIDE_LEFT_RIGHT
Action situation: 9 -> I stopped
Motor system situation 9 get decision for DECIDE_LEFT_RIGHT
Recognized place : F and decided to CHOOSE_LEFT
```

Figure 7.8. Part of the decision making process. Shows how location F gets detected and the decision is made to first stop and then turn left.

7.5 Activation of system 2

The model was created in such a way that system 1 would always be active, but that system 2 would only be activated when necessary. A neural network that uses sonar for it's input detects these kind of situations. One of the main arguments theory states why we have evolved such a distinction is efficiency. system 2 can make more elaborate decisions, but comes with the cost of using much more energy. The same thing is true for the robot. In the case of walking a sequence, constantly activating system 2 would also mean that the system would constantly activate the location detection system, which is an even more resource heavy system than the sonar based situation recognizing system. Constant activation of this network would dramatically decrease the operating speed of the robot. Practice runs already show that the robot slows down at locations where system 2 should activate and the robot walks faster when this system is deactivated.

Data from the sequence walk is gathered and put together in an image that shows at which moment in time system 2 was activated and when it was deactivated again. Figure 7.9 displays the activation over time. To get a good picture of the moments system 2 was activated, the location numbers are also plotted below the x-axis. These numbers are the same situation numbers which were used on the tracking image and in the decision making transcript.

It can clearly be seen that system 2 is only activated briefly each time and the moments of activation correspond to the situation numbers. You can also see that it keeps disabled when the robot walks through long hallways. Activation does occur at locations which would only need

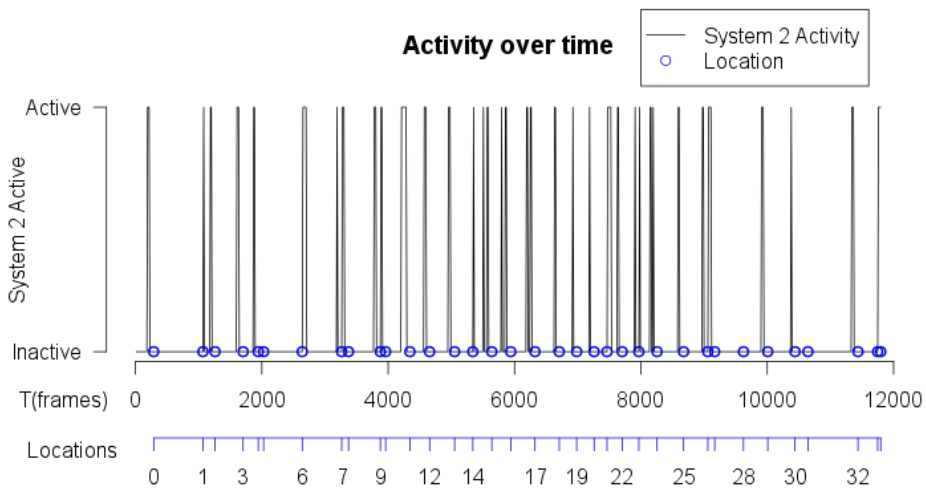

Figure 7.9. System 2 activity over time. System two activation is plotted against frame count. The number below the frame count correspond to the numbers that were placed on the robot tracking map and the numbers in the recording of the decision making process. The position of the numbers under the graph correspond to the moments in time these situations were visited.

system 1 activation, but this can probably be explained by the picture not being clear enough yet and system 2 becomes activated. For instance it shows system 2 activity right before it reaches situation 1, which is a left turn that should be handled by system 1 alone.

8

Conclusion and Discussion

Results from the different analysis methods are discussed in this chapter. From the PCA analysis to the lesion studies and functioning of the model.After that a discussion is started about the relevance of this kind of research for science.

8.1 PCA Analysis

The most important conclusion that can be made from the PCA analysis is that representations can be formed at multiple levels of processing. In the case of the hallway the representation is already formed in the S4 layer and might even be present in the S2 or even S1 layer. Nothing changes to this representation when information is passed through the IT layer.

Other situations like the crossing have a strong IT layer component, but no single strong S4 component. Situations like the t-junction only have a single representation in the output layer. These results indicate that some situations are more easy to learn than others.

A similar conclusion can be drawn from the PCA analysis of the location detection network. There are no direct representations of locations in the IT layer, but there are strong components in the S4 and V4 layers. It looks like the IT layer has learned to combine these components. The activity of a single output neuron is then the representation of a location. It is not a big surprise that for the detection of locations all the processing layers are needed to form representations. Locations contain a lot of information and because of their uniqueness are very complex objects to learn.

These findings are all much in line with the literature which states that when information

passes through the ventral stream, the neurons in the different areas can can encode for increasing complex structures. From simple bars in the V1 to complete spatial invariant representations in the temporal lobe. Even the activity of just a single neuron can be the abstract representation of a complex object as research by Quiroga, Reddy, Kreiman, Koch, and Fried (2005) shows. They did single cell recordings in human subjects on neurons in the Medial Temporal Lobe area, in the area that is known for recognizing faces and found that just a single neuron started firing when the subjects were shown a picture of Jennifer Aniston.

8.2 Lesion Studies

The lesion studies clearly show that lesioning one of the networks layers impairs it's ability to recognize locations. The color blindness effects occurred mostly as expected, although some locations were correctly identified where one would not expect it. Maybe other information was more important there.

The dramatic effect of lesioning the sonar input layer was unexpected, because the networks with and without sonar performed almost the same in the virtual world. Principal component analysis reveals why such a dramatic effect was present. The sonar and visual information both exert a strong influence on the final result and removing one of the two information streams almost completely removes the ability to recognize locations.

Just lesioning one the color channels doesn't remove all the visual information so the detection of locations is only impaired to a lesser extend. Lesioning the complete visual or sonar system can be compared to a situation where a seeing person is blindfolded and then asked to recognise locations. He will probably fail miserably, although there might be some places he recognises because of other features like known sounds or smells.

8.3 Model

The robot is able to navigate it's way through the virtual world so it is safe to say that the model has produced the output it was supposed to. The next section will review in more detail how well the different parts of the model have functioned when it was put to the test by performing the route navigation task.

8.3.1 Dual Process

The distinction between system 1 and system 2 works as intended. As the time series plot and the transcript show, system 2 is disabled most of the time. All the left turns, right turns, dead ends hallways were controlled by system 1 and all the more complex decisions were made by system 2. This means that the separate early attention mechanism did his job of detecting situations that need more elaborate processing.

The results did show that there were moments at which system 2 didn't need to be activated, but this was probably due tot the picture not being clear enough for the network to come to a correct decision.

The motor system is the place where the two systems come together and have their common pathway to overt behaviour. System 2 is then always favoured above System 1. In humans the battle between the two systems is much more complex. In humans it depends on the strengths of habits and the amount of energy available for system 2 to override these habits. Strong habits require more energy to overcome than weaker ones. There is ongoing research on this subject. Insight in dual-processing is especially useful in treatment of addiction. Addiction is a situation where unhealthy habits have become so strong that it is nearly impossible for system 2 to prevent this addictive behaviour. Much research on addiction focusses on restoring the balance between the two processes (Hofmann, Friese, & Wiers, 2008; Wiers, Eberl, Rinck, Becker, & Lindenmeyer, 2011; Gollwitzer, Fujita, & Oettingen, 2004).

The most important advantage such a distinction between the two systems has for the robot is energy efficiency. The robot's slows down when system 2 is activated, indicating a much higher need for processing power. Humans benefit in a similar way. Only having a system 2 type of decisioning system would probably need more energy than a human can produce. Because of this distinction humans can make use of the automatic system for easy habit like decisions. It then saves energy for decisions that need more elaboration. Without this distinction we probably wouldn't be able to have such high levels of cognitive processing. Even if we did we wouldn't have the time to use it, since we would be eating all day just to provide the brain with the huge amounts of energy it would need.

As Strack and Deutsch (2004) already pointed out there needs to be a mechanism to active the system 2 brain areas. The structure that was built in section 5.2.2 does a good job in activating these areas. Since it is basically a simplified version of the ventral stream model, it is not unthinkable that similar structures can exist in living organisms.

8.3.2 Motor system

In this systems decisions are translated into actions. The two inhibition mechanisms do a good job of preventing unwanted actions. The first inhibition mechanism decides whether it is possible to perform an action or not. It prevented the robot from crashing, wrong decisions were made but this system prevented them from being turned into actions. It is not hard to see why any creature would benefit from such a system. Not having such a system could result in situations like turning left in a car immediately after your navigation device tells you to and then finding yourself floating in a river because you had to drive 50 meters and use the bridge to get across.

There are some unwanted actions which can get past the first inhibition mechanism and can cause unwanted interference with an action that is already being executed. Luckily the second inhibition mechanism prevents that from happening.

8.4 Discussion

When all the parts of the model are connected a complete system emerges which uses a model of the ventral stream for recognition, has a dual-process model for decision making, an early attention mechanism for allocation extra resources and a pre-frontal cortex/basal ganglia like motor system. Although in no way it can be said that this resembles an animal brain structure it does show that when you combine the basic properties of what is known about these structures in animals, it is possible to let a virtual creature perform the complex task of learning a route based on recognition. All of the different parts are just ”simple” objects which can only perform a single task, but combined they become a powerful and flexible structure.

The situation detection network based on sonar creates spatial invariant representations of the different situations in the virtual world much like the human ventral stream does. You cannot say it has learned it in exactly the same way a real brain would do, but the end result is similar. Because of the way the experiment was set up, data could be gathered to follow the decision making process. Also neuronal activation could be exported to data files for statistical analyses. This level of detail is not possible in human or animal studies which can only be studied empirically.

For example just asking the robot which location it detects and what it’s next action will be could lead to the following:

```
Recognized place : A and decided to CROSS_STRAIGHT
Recognized place : I and decided to CHOOSE_LEFT
Recognized place : H and decided to CROSS_STRAIGHT
Recognized place : G and decided to CHOOSE_LEFT
Recognized place : G and decided to CROSS_STRAIGHT
Recognized place : F and decided to CHOOSE_LEFT
Recognized place : C and decided to CHOOSE_RIGHT
Recognized place : D and decided to CHOOSE_LEFT
```

Figure 8.1. Part of the decision making process. This only shows which locations were recognized and the decision that was made at each location.

Asking a human subject where it is and which direction to take would lead to exactly the same information. When encoded in the same way there would be no way to tell the robot and human apart. Thus simulations can lead to the same behavioural data. The main difference is that with a simulation it is also possible to study in much more detail how the outcomes were produced. Naturally having the same outcome doesn't mean that the processes leading to the result are the same. Since this a CCN model and structures are created according to the CCN ideals, especially the neuroscience ideal, the output produced by the neural networks should be at least biologically plausible.

All the neuronal models are based on the layered organization of the ventral stream. This layered organization is not just seen in the ventral stream, but is present in the whole cortex. Neurons are organized in horizontal layers and vertical columns. Many of these structures operate in a hierarchical way through pathways, like the ventral stream. This suggests that the cognitive processes in the brain are implemented with networks and systems based on uniform structures (van der Velde, 2010).

The structures used in this thesis point in a similar direction. The networks are all made up out of feed forward layers. These in turn are made up out of one or more groups of neurons. These layers start as uniform building blocks, but become specialized after training. The different networks used in this thesis show that they can use different kinds of input and produce totally different output. For instance it uses sonar for learning situations, but also for detecting whether it is the right moment for action. It can also use vision or even a combination of vision and sonar to detect locations. These kind of layered structures provide a very powerful way of learning complex situation or objects.

8.4.1 Relevance for science

There is an ongoing debate about the usefullness of using an animat or ”iguana” approach for science. This debate focusses mainly on the work of Randall Beer. He is building minimally cognitive systems and places them in a created environment where they learn to discriminate between diamonds and circles. Beer sees this approach as a new way of understanding cognition in general (Beer, 1996; Beer, 2008).

It is this generalization that is mainly criticized and most strongly by Webb (2009). She states that work on animats can only be considered as the exploration of artificial systems. For biological relevance you need real iguanas and not the three-wheeled Martian ones. This vision is much shared by Huelse and Lee (2010) which also states that a robotic experiment can not provide new scientific knowledge about a biological phenomenon. They give four qualities by which this approach might be of indirect value for science.

- Biologically inspired implementations
- Proof of concept
- Demonstrations of general principles
- Support of the development of new formalisms, paradigms, and tools

Noble and de Pinedo (2009) agree with Webb on that a model instantiated as a computer program is and only will ever be a computer program, but they cannot agree with Webb on her critique of Beer’s work. For them using specialised abstract models does seem a legitimate new way of understanding cognition. Fitzpatrick (n.d.) agrees with stating that we might find general principles far more rapidly by invention than by the horrendously difficult task of reverse-engineering nature.

Beer and Williams (2009) responded to Webb’s critique. Beer states that Webb tries to enforce a specific modelling methodology he calls data-driven modelling. By which he means trying to model data that has been gathered from real animals and then use the model for validating experiments done on the real animal or by doing predictions that can then be tested on the real animal.

Beer calls its own approach theory-driven modelling. He states that these kind of models are not specific to any particular existing system, but rather their target is the set of such systems that exhibit some phenomena of interest. The models try to capture the essential features that

are common to all members of this set. Beer quotes Carlip (2003) about reconciling quantum mechanics with general relativity to illustrate the validity of such models for science.

> Faced with such problems, it is natural to look for simpler models that share the important conceptual features of general relativity ... General relativity in 2+1 dimensions – two dimensions of space and one of time – is one such model.... With a few exceptions, (2+1)-dimensional solutions are physically quite different from those in 3+1 dimensions, and the (2+1)-dimensional model is not very helpful for understanding the dynamics of realistic quantum gravity. But for the analysis of conceptual problems – the nature of time, the construction of states and observables, the role of topology and topology change, the relationship among different approaches to quantization – the model has proven highly instructive. (Carlip, 2003).

van der Velde (2010) sees an important role for computational cognitive neuroscience. He states that for the first time in history it is possible to investigate the neural mechanisms that produce human cognition. He mentions that we now have the right research methods and techniques, have the theoretical knowledge and the computer power to perform large scale simulations. He also envisions computational cognitive neuroscience to play a fundamental role there by combining the different lines of research needed to understand the complexity of the brain and the cognitive processes it produces.

This thesis shares much of this vision. Performing computer simulations gives a good extension of the more traditional methods. It provides a bottom-up approach which would otherwise not be available. The small scale of the simulations performed in this research, just some 10 thousands of neurons compared to the billions in the human, already gives a clue of what would be possible when really large scale simulations are possible. Models can easily be adjusted and analysis can be done at a level of detail which will never be possible in human subjects.

Although computer power is rapidly increasing it still is the major bottleneck for this kind of research. The simulations done in this theses really pushed the limits of the hardware it operated on. Processing speeds are nowhere near what a brain can do. The major issue here is that although normal computer processors can perform calculations at a very high speed, they outperform a single neuron easily, they can only do that one calculation at a time. Even though a neuron is much slower there are billions of them operating in parallel in the human

brain. Recent advances in parallel computing might help to overcome this problem. The Neurogrid project for example can already operate a neural network consisting of 1 million neurons in real-time and with a power consumption of just 5 watts. For more information see their website http://www.stanford.edu/group/brainsinsilicon/index.html. In another interesting study researchers from Japan are already trying to create parallel processors on a molecular scale (Bandyopadhyay, Pati, Sahu, Peper, & Fujita, 2011).

These recent advances and the important role cognitive computational neuroscience can play in uncovering the neural mechanisms of cognition, makes this relatively new field of research a very interesting one which might have a very bright future, one in which Markam's dream might become reality after all.

Bibliography

Aisa, B., Mingus, B., & O'Reilly, R. (2008, October). The emergent neural modeling system. *Neural Networks*, *21*(8), 1146–1152. doi:10.1016/j.neunet.2008.06.016

Anzai, A., Peng, X., & Van Essen, D. C. (2007, September). Neurons in monkey visual area v2 encode combinations of orientations. *Nature Neuroscience*, *10*(10), 1313–1321. doi:10.1038/nn1975

Ashby, F. G. & Helie, S. (2011). A tutorial on computational cognitive neuroscience: modeling the neurodynamics of cognition. *Journal of Mathematical Psychology*, *55*(4), 273–289. doi:10.1016/j.jmp.2011.04.003

Bandyopadhyay, A., Pati, R., Sahu, S., Peper, F., & Fujita, D. (2011). Massively parallel computing on an organic molecular layer. *CoRR*, *abs/1110.5844*.

Beer, R. D. (1996). Toward the Evolution of Dynamical Neural Networks for Minimally Cognitive Behavior. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.7898

Beer, R. D. (2008). The dynamics of brain-body-environment systems: a status report. *Handbook of cognitive science: An embodied approach*, 99–120.

Beer, R. D. & Williams, P. L. (2009, August). Animals and animats: why not both iguanas? *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, *17*(4), 296–302. doi:10.1177/1059712309340844

Benkő, A. & Lányi, C. (2009). History of artificial intelligence. In *Encyclopedia of information science and technology* (pp. 1759–1762). IGI Global.

Buchanan, B. (2005). A (very) brief history of artificial intelligence. *AI Magazine*, *26*(4), 53.

Carlip, S. (2003). *Quantum gravity in 2+ 1 dimensions*. Cambridge University Press.

Cattell, R. & Parker, A. (2012). *Challenges for brain emulation: why is building a brain so difficult*. Retrieved from http://synapticlink.org/Brain%20Emulation%20Challenges.pdf

Collignon, O., Lassonde, M., Lepore, F., Bastien, D., & Veraart, C. (2007). Functional cerebral reorganization for auditory spatial processing and auditory substitution of vision in early blind subjects. *Cerebral Cortex*, *17*(2), 457–465. doi:10.1093/cercor/bhj162. eprint: http://cercor.oxfordjournals.org/content/17/2/457.full.pdf+html

Collignon, O., Vandewalle, G., Voss, P., Albouy, G., Charbonneau, G., Lassonde, M., & Lepore, F. (2011). Functional specialization for auditoryspatial processing in the occipital cortex of congenitally blind humans. *Proceedings of the National Academy of Sciences*, *108*(11), 4435–4440. doi:10.1073/pnas.1013928108. eprint: http://www.pnas.org/content/108/11/4435.full.pdf+html

Corbetta, M., Patel, G., & Shulman, G. L. (2008). The reorienting system of the human brain: from environment to theory of mind. *Neuron*, *58*(3), 306–324. doi:10.1016/j.neuron.2008.04.017

Crick, F. (1989, January). The recent excitement about neural networks. *Nature*, *337*(6203), pages. Retrieved from http://ukpmc.ac.uk/abstract/MED/2911347

De Schutter, E. (2008, May). Why are computational neuroscience and systems biology so separate? *PLoS Comput Biol*, *4*(5), e1000078. doi:10.1371/journal.pcbi.1000078

de Garis, H., Shuo, C., Goertzel, B., & Ruiting, L. (2010). A world survey of artificial brain projects, part i: large-scale brain simulations. *Neurocomputing*, *74*, 3–29. doi:10.1016/j.neucom.2010.08.004

Dennett, D. C. (1978, March). Why not the whole iguana? *Behavioral and Brain Sciences*, *1*(01), 103. doi:10.1017/S0140525X00059859

DiCarlo, J. J. & Cox, D. D. (2007, August). Untangling invariant object recognition. *Trends in Cognitive Sciences*, *11*(8), 333–341. doi:10.1016/j.tics.2007.06.010

Djurfeldt, M., Ekeberg, Ö., & Lansner, A. (2008). Large-scale modeling - a tool for conquering the complexity of the brain. *Frontiers in Neuroinformatics*, *2*. Retrieved from http://www.frontiersin.org/Journal/Abstract.aspx?s=752&name=neuroinformatics&ART_DOI=10.3389/neuro.11.001.2008

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211. doi:http://dx.doi.org/10.1016/0364-0213(90)90002-E

Evans, J. (n.d.). Dual system theories of cognition. some issues. Retrieved from http://csjarchive.cogsci.rpi.edu/proceedings/2006/docs/p202.pdf

Evans, J. S. (2008). Dual-processing accounts of reasoning, judgment, and social cognition. *Annual Review of Psychology*, *59*(1), 255–278. doi:10.1146/annurev.psych.59.103006.093629

Evans, J. S. (2003). In two minds: dual-process accounts of reasoning. *Trends in Cognitive Sciences*, *7*(10), 454–459. doi:10.1016/j.tics.2003.08.012

Fitzpatrick, P. (n.d.). Anatomy of a martian iguana. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.6656&rep=rep1&type=pdf

Gollwitzer, P. M., Fujita, K., & Oettingen, G. (2004). *Planning and the implementation of goals*. Bibliothek der Universität Konstanz.

Gross, C. et al., Rocha-Miranda, C. et al., Bender, D., et al. (1972). Visual properties of neurons in inferotemporal cortex of the macaque. *Journal of neurophysiology*, *35*(1), 96.

Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley.

Hodgkin, A. & Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, *117*(4), 500.

Hofmann, W., Friese, M., & Wiers, R. W. (2008). Impulsive versus reflective influences on health behavior: a theoretical framework and empirical review. *Health Psychology Review*, *2*(2), 111–137. doi:10.1080/17437190802617668. eprint: http://www.tandfonline.com/doi/pdf/10.1080/17437190802617668

Hubel, D. H. & Wiesel, T. N. (1998, March). Early exploration of the visual cortex. *Neuron*, *20*(3), 401–412. doi:10.1016/S0896-6273(00)80984-8

Huelse, M. & Lee, M. (2010). The logic of biological inspired robotics. In *Proceedings of the international symposium on ai-inspired biology, chappell, j., thorpe, s., hawes, n., sloman, a.(eds.) at the aisb* (pp. 43–50). Retrieved from http://cadair.aber.ac.uk/dspace/handle/2160/22/simple-search?location=2160/22&query=Biologically+Inspired+Robotics+and+Controlsort_by=0

Kalat, J. W. (2007). *Biological psychology*. Wadsworth Publishing Company.

Kim, Y. & Lewis, F. (1999, April). Neural network output feedback control of robot manipulators. *Robotics and Automation, IEEE Transactions on*, *15*(2), 301–309. doi:10.1109/70.760351

Lamme, V. A., Supèr, H., & Spekreijse, H. (1998). Feedforward, horizontal, and feedback processing in the visual cortex. *Current Opinion in Neurobiology*, *8*(4), 529–535. doi:10.1016/S0959-4388(98)80042-1

Mahon, B. Z. & Caramazza, A. (2011, March). What drives the organization of object knowledge in the brain? *Trends in Cognitive Sciences*, *15*(3), 97–103. doi:10.1016/j.tics.2011.01.004

Markram, H. (2012). The human brain project. *Scientific American*, *306*(6), 50–55. doi:10.1038/scientificamerican0612-50

McCulloch, W. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, *5*, 115–133. 10.1007/BF02478259. Retrieved from http://dx.doi.org/10.1007/BF02478259

Meyer, J.-A. (1996). Artificial life and the animat approach to artificial intelligence. Academic Press. Retrieved from http://www.isir.upmc.fr/files/1996COS444.pdf

Mishkin, M., Ungerleider, L. G., & Macko, K. A. (1983). Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, *6*, 414–417. doi:10.1016/0166-2236(83)90190-X

Noble, J. & de Pinedo, M. (2009, July). Iguana modeling is not the only game in town. *Adaptive Behavior*, *17*(4), 331–333. doi:10.1177/1059712309340856

Oram, M. W. & Perrett, D. I. (1994). Modeling visual recognition from neurobiological constraints. *Neural Networks*, *7*, 945–972. ¡ce:title¿Models of Neurodynamics and Behavior¡/ce:title¿. doi:http://dx.doi.org/10.1016/S0893-6080(05)80153-4

O'Reilly, R. C., Munakata, Y., Frank, M. J., Hazy, T. E., & Contributors. (2012). *Computational cognitive neuroscience*. Wiki Book, 1st Edition, URL: http://ccnbook.colorado.edu. Retrieved from http://ccnbook.colorado.edu

Ou, C.-M., Li, H.-Y., & Hwang, W.-J. (2012, December). Efficient k-winner-take-all competitive learning hardware architecture for on-chip learning. *Sensors*, *12*(12), 11661–11683. doi:10.3390/s120911661

Peissig, J. J. & Tarr, M. J. (2007, January). Visual object recognition: do we know more now than we did 20 years ago? *Annual Review of Psychology*, *58*(1), 75–96. doi:10.1146/annurev.psych.58.102904.190114

Petrov, A. A., Jilk, D. J., & O'Reilly, R. C. (2010). The leabra architecture: specialization without modularity. *Behavioral and Brain Sciences*, *33*(04), 286–287. doi:10.1017/S0140525X10001160. eprint: http://journals.cambridge.org/article_S0140525X10001160

Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., & Fried, I. (2005, June). Invariant visual representation by single neurons in the human brain. *Nature*, *435*(7045), 1102–1107. Retrieved from http://dx.doi.org/10.1038/nature03687

R Core Team. (2013). *R: a language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. Retrieved from http://www.R-project.org/

Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton project para*. Cornell Aeronautical Laboratory.

Rust, N. C. & DiCarlo, J. J. (2010, September). Selectivity and tolerance (invariance) both increase as visual information propagates from cortical area v4 to it. *Journal of Neuroscience*, *30*(39), 12978–12995. doi:10.1523/JNEUROSCI.0179-10.2010

Sejnowski, T., Koch, C., & Churchland, P. (1988). Computational neuroscience. *Science*, *241*(4871), 1299–1306. doi:10.1126/science.3045969. eprint: http://www.sciencemag.org/content/241/4871/1299.full.pdf

Strack, F. & Deutsch, R. (2004). Reflective and impulsive determinants of social behavior. *Personality and Social Psychology Review*, *8*(3), 220–247. doi:10.1207/s15327957pspr0803_1. eprint: http://psr.sagepub.com/content/8/3/220.full.pdf+html

Tanaka, K. (1996). Representation of visual features of objects in the inferotemporal cortex. *Neural Networks*, *9*(8), 1459–1475. ¡ce:title¿Four Major Hypotheses in Neuroscience¡/ce:title¿. doi:http://dx.doi.org/10.1016/S0893-6080(96)00045-7

Thaler, L., Arnott, S. R., & Goodale, M. A. (2011, May). Neural correlates of natural human echolocation in early and late blind echolocation experts. *PLoS ONE*, *6*(5), e20162. doi:10.1371/journal.pone.0020162

Theeuwes, J. (2010, October). Top-down and bottom-up control of visual selection. *Acta Psychologica*, *135*(2), 77–99. doi:10.1016/j.actpsy.2010.02.006

Toates, F. (2006). A model of the hierarchy of behaviour, cognition, and consciousness. *Consciousness and Cognition*, *15*(1), 75–118. doi:10.1016/j.concog.2005.04.008

Ungerleider, L. G. & Haxby, J. V. (1994). What and where in the human brain. *Current Opinion in Neurobiology*, *4*(2), 157–165. doi:10.1016/0959-4388(94)90066-3

van der Velde, F. (2010, January). Where artificial intelligence and neuroscience meet: the search for grounded architectures of cognition. *Adv. in Artif. Intell. 2010*, 5:3–5:3. doi:10.1155/2010/918062

van der Velde, F. & de Kamps, M. (2001, May). From knowing what to knowing where: modeling object-based attention with feedback disinhibition of activation. *Journal of Cognitive Neuroscience*, *13*(4), 479–491. doi:10.1162/08989290152001907

Vermeulen, W., van der Wolk, P., de Weijer, A., & van der Zwaag, S. (1996). Prediction of jominy hardness profiles of steels using artificial neural networks. *Journal of Materials Engineering and Performance*, *5*, 57–63. 10.1007/BF02647270. Retrieved from http://dx.doi.org/10.1007/BF02647270

Webb, B. (2009, July). Animals versus animats: or why not model the real iguana? *Adaptive Behavior*, *17*(4), 269–286. doi:10.1177/1059712309339867

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences* (Doctoral dissertation, Harvard University, Cambridge, MA).

Werbos, P. J. (2009, April). 2009 special issue: intelligence in the brain: a theory of how it works and how to build it. *Neural Netw. 22*(3), 200–212. doi:10.1016/j.neunet.2009.03.012

Werbos, P. (1990, October). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550–1560. doi:10.1109/5.58337

Wiers, R. W., Eberl, C., Rinck, M., Becker, E. S., & Lindenmeyer, J. (2011). Retraining automatic action tendencies changes alcoholic patients approach bias for alcohol and improves treatment outcome. *Psychological Science*, *22*(4), 490–497. doi:10.1177/0956797611400615. eprint: http://pss.sagepub.com/content/22/4/490.full.pdf+html

Zhang, Y. & Wu, L. (2009). Stock market prediction of s&p 500 via combination of improved bco approach and bp neural network. *Expert Systems with Applications*, *36*(5), 8849–8854. doi:10.1016/j.eswa.2008.11.028

A

Leabra training algorithm

The most widely used training algorithm backpropagation has a big disadvantage when applied to CCN models. It is said to be very biologically implausible. Crick (1989) mentions that backpropagation is unrealistic in almost every way. One important thing he mentions is that it would require the rapid transmission of information backwards along the axon. He thinks it is highly unlikely that the brain behaves in such a way.

Leabra is an acronym for ”Local, Error-driven and Associative, Biologically Realistic Algorithm”. This implies that this algorithm aims to be more biologically plausible. Leabra does this by combining two learning methods, error driven and self-organizing (”Hebbian”) learning. This in contrary to backpropagation which only uses error-driven learning. To make error-driven learning more biologic realistic Leabra uses bidirectional connectivity between neurons instead of moving information backwards. It is also unique because it features a k-winner-takes-all (kWTA) inhibitory function. This function creates inhibitory competition among units within a layer. It does this by computing a uniform level of inhibitory current for all units in the layer, such that the $k + 1$th most excited unit within a layer is below its firing threshold, while the kth is above threshold.

Error driven learning takes place by comparing outcome with expectation. The period in which the network creates it’s expectation is called the minus phase and the period in which the network observes the outcome is called the plus phase. The difference between these phases is the error. This error can than be propagated into the network through the bidirectional connections. This in contrary to the backpropagation algorithm, which also calculates the difference between expecattion and outcome, but propagates this error back through it’s

connections.

When all the features of Leabra are combined it should be highly compatible with the known biological features of the neocortex (O'Reilly, Munakata, Frank, Hazy, & Contributors, 2012; Ou, Li, & Hwang, 2012).

Leabra is then much more in line with the first CCN ideal, the neuroscience ideal, than backpropagation. Since Emergent uses Leabra as it's default learning mechanism, Leabra is clearly the best learning algorithm to choose for this research.

B

Programming code

In this appendix some java code samples are shown. These samples come from the java project that were created for building the robot and its virtual world and implementing some parts of the model.

B.1 Detected objects

```
public enum DetectedSonarObjects {
    HALLWAY,
    LEFT_TURN,
    RIGHT_TURN,
    DEAD_END,
    FRONT_LEFT,
    FRONT_RIGHT,
    T_JUNCTION,
    CROSSING,
    NONE,
}
```

Listing 1: Java enum structure used to represent the detected objects. These are the direct representations of the output neurons from the sonar detection network.

B.2 Creating the robot and the virtual world

```
public WayfinderRobot(Vector3d position, String name) {
        super(position, name);
        this.name = name;
        robotListeners = new ArrayList<RobotListener>();
        sonars = RobotFactory.addSonarBeltSensor(this, 24);
        cameraFront = RobotFactory.addCameraSensor(this);
        cameraLeft = RobotFactory.addCameraSensor(this);
        cameraLeft.rotateY(Math.PI / 2);
        cameraRight = RobotFactory.addCameraSensor(this);
        cameraRight.rotateY(-Math.PI / 2);
        cameraRight.rotateY(-Math.PI/4);
    }
```

Listing 2: Part of the robot constructor. The code shows how a new instantion of the robot will be given 24 sonar sensors and three camera's. A camera on the left, front and right side of the body.

```
public TestWorld() {
        testWorld = this;
        worldSize = 30;
        Wall w = new Wall(new Vector3d(-15f, 0, 13.75f), 2.5f, 2, this);
                //Rotate and give it a green color
        w.rotate90(1);
        w.setColor(new Color3f(Color.green));
        Wall w1 = new Wall(new Vector3d(0, 0, 15), 30, 2, this);
                //Create more walls
                add(w); //Add the wall to the world
        }
```

Listing 3: This code shows a part of the code that is used as constructor to create a new virtual environment. It shows the creation and positioning of walls in the world.

B.3 Model implementation

```java
/**
* Determines the next action to perform, based on Objects input. There are
* some actions that don't need any special attention. These are taking a
* left and right turn, walking through a hallway and turning around
* when reaching a dead end.
*
* @param object The detected object
*/
private void determineNextAction(Enum object, Modality modality) {
 int location = MotorSystem.locationCounter;
 if (object instanceof DetectedSonarObjects) {
     DetectedSonarObjects detectedObject = (DetectedSonarObjects) object;
     if (!detectedObject.equals(previousObj)) {
        switch (detectedObject) {
            case HALLWAY:
                OutputFrame.appendLine("System: 1 location: " + location
                + " -> Move forward");
                decision = Decisions.MOVE_FORWARD;
                break;
            case LEFT_TURN:
                OutputFrame.appendLine("System: 1 location: " + location
                + " -> Left turn");
                decision = Decisions.TURN_LEFT;
                break;
            case RIGHT_TURN:
                OutputFrame.appendLine("System: 1 location: " + location
                + " -> Right turn");
                decision = Decisions.TURN_RIGHT;
                break;
            case DEAD_END:
                OutputFrame.appendLine("System: 1 location: " + location
                + " -> Dead end, turn around");
                decision = Decisions.GO_BACK;
                break;
            default:
                OutputFrame.appendLine("System: 1 location: " + location
                + " -> "+detectedObject);
                decision = Decisions.NO_DECISION;
                break;
        }
    }
    previousObj = (DetectedSonarObjects) object;
}
```

Listing 4: System 1 associations between recognised objects and actions.