10.2 Robot results

In the following scale and speed correspond to the real settings.

The settings for the following were the four mazes from which one or several logs had been drawn, and six extra (invented) mazes used for testing purposes.

- The performances were not better in the "known" mazes than in the invented mazes, showing that the strategies had really been abstracted from their original settings.
- All tables had been explored after at most 11 minutes by robot_1. After 10 minutes robot_2 had explored all tables in all but one maze. That last maze had an isolated central table which robot_2 often bypassed in its exploration of the empty space.
- On average, 83% of the tables had been explored after 3 minutes by robot_1 and 86% by robot_2.
- Dividing the ground in squares 20 pixels across[12], which corresponds to the average "width" of a subject as seen on the videos, between 78% and 94% of all reachable squares (ground and tables) had been reached at least once after 10 minutes by robot_1, and between 80% and 94% by robot_2, the actual average values varying according to maze size and complexity. The improvement mostly happened in complex mazes with many tables.
- These percentages increase with run duration.

The difference between the efficiency of robot_1 and of robot_2 might seem slight, but as performance improves each percent point becomes harder to gain than the previous one. So the difference between 83% and 86% of the tables is more significant than would have been a difference between 63% and 66% of the tables.

11 Conclusion

Closing the loop: As our robots move about in the mazes, their observable actions can be recorded. Thus the teacher/learner/teacher loop, sometimes also refered to as the raw-data/trained-system/raw data (or raw-data/learned-strategies/raw-data) loop, is closed because new logs can be generated and learning can be achieved from these new logs. These new logs are neither a better nor a worse model than the originals, were the robot generating them to run for a sufficiently long time and in a sufficiently complex maze for the randomness to be overcome by statistical significance they would amount to the same information.

H-CogAff validation: H-CogAff describes the functions of the mind [22]. Our experiments only validate the H-CogAff model in a severely limited and constrained context. They also validate the global alarm system, with the same restrictions and more because a single variable was used to trigger strategy changes, and this variable was neither learnt nor even statistically determined but was chosen by empirical reasoning. We can however say that an instant reaction which resets every control variable upon a bump improves performance, and that this seems to indicate that simulating a global alarm system, hopefully in more advanced ways, in other robotic controllers could be a promising line of research.

ACKNOWLEDGEMENTS

Our link between the global alarm system and emotions was first suggested by David A. Focil.

[12] We were given the plans of the mazes by the psychologists, but these plans had no scale. When we inquired about it we were told that the corresponding rooms still existed and could be measured. We declined and used pixels for our distance unit.

REFERENCES

[1] Jean-Francois Richard Aldo Zanga and Charles Tijus, 'Implicit learning in rule induction and problem solving', *Thinking and Reasoning*, **10 (1)**, 55–83, (2004).

[2] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn, 'Learning to imitate corresponding actions across dissimilar embodiments', *IEEE Transactions on Systems, Man, and Cybernetics*, **32**, 482–496, (2002).

[3] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn, 'Correspondence mapping induced state and action metrics for robotic imitation', *Special Issue on Robot Learning by Observation, Demonstration and Imitation*, (2006).

[4] Billard and Hayes, 'Drama, a connectionist architecture for control and learning in autonomous robots', *Adaptive Behaviour*, **7**, 35–64, (1999).

[5] A. Billard and R. Siegwart, 'Robot learning from demonstration', *Robotics and Autonomous Systems*, **47**, 65–67, (2004).

[6] S. Calinon and A. Billard, *Imitation and Social Learning in robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, Cambridge University Press, 2007.

[7] S. Calinon, F. Guenter, and A. Billard, 'On learning, representing and generalising a task in a humanoid robot', *IEEE Transactions on Systems, Man and Cybernetics, Part B, Special Issue on Robot Learning by Observation, Demonstration and Imitation*, **37 (2)**, (2007).

[8] G. E. Hinton D. E. Rumelhart and R. J. Williams, 'Learning internal representations by error propagation', *David E. Rumelhart and James A. McClelland*, **1**, (1986).

[9] Demiris and Hayes, *Imitation in animals and artifacts*, 2001.

[10] R. Dillmann, 'Teaching and learning of robot tasks via observation of human performance', *Robotics and Autonomous Systems*, **47**, 109–116, (2004).

[11] M. Felkin, 'Learning by observation and induction the strategies of humans placed in a problem-solving context', *PhD Thesis*, (2008).

[12] M. Felkin and Y. Kodratoff, 'High level imitation learning in a treasure hunting task', *Proceedings of Plan, Activity, and Intent Recognition (PAIR) workshop, International Joint Conferences on Artificial Intelligence (IJCAI 2009)*, (2009).

[13] Valentina Iemmi, 'Les apports des nouvelles technologies a la psychologie clinique: Les robots comme compagnons therapeutiques', (2005).

[14] Heumer G. Jung B., Ben Amor H. and Weber M., 'From motion capture to action capture: A review of imitation learning techniques and their application to vr-based character animation', *Proceedings ACM VRST*, (2006).

[15] Henri Kraus, 'A formal theory of plan recognition and its implementation', *Reasoning About Plans, by J.F. Allen, H.A. Kautz, R.N. Pelavin, and J.D. Tennenberg*, 69–126, (1991).

[16] J. P. Müller, 'A conceptual model of agent interaction', *Second International Working Conference on Cooperating Knowledge Based Systems (CKBS-94)*, 389–404, (1994).

[17] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

[18] J. R. Quinlan, 'Improved use of continuous attributes in c4.5', *Journal of Artificial Intelligence Research*, **4**, 77–90, (1996).

[19] S. Schaal, A. Ijspeert, and A. Billard, 'Computational approaches to motor learning by imitation', *Philosophical Transactions of the Royal Society of London: series B, Biological Sciences*, **358**, 537–547, (2003).

[20] Storz J. Shon A. and Rao R., 'Towards a real-time bayesian imitation system for a humanoid robot', *Proceedings of the IEEE International Conference on robotics and Automation*, 2847–2852, (2007).

[21] A. Sloman, 'Varieties of affect and the cogaff architecture schema', *Proceedings Symposium on Emotion, Cognition, and Affective Computing AISB'01 Convention*, 39–48, (2001).

[22] Aaron Sloman, 'Some requirements for human-like robots: Why the recent over-emphasis on embodiment has held up progress', *Creating Brain-Like Intelligence, LNAI 5436*, 248–277, (2009).

[23] C. Tijus, N. Bredeche, Y. Kodratoff, M. Felkin, C. Hartland, V. Besson, and E. Zietti, 'Human heuristics for a team of mobile robots', *Proceedings of the 5th International Conference on Research, Innovation and Vision for the Future (RIVF'07)*, (2007).

[24] Mark P. Woodward and Robert J. Wood, 'Using bayesian inference to learn high-level tasks from a human teacher', *International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09)*, (2009).

[25] Victor Lesser Xiaoqin Zhang and Tom Wagner, 'A layered approach to complex negotiations', **2(2)**, 91–104, (2004).

Proceedings of the International Symposium on AI-Inspired Biology, Jackie Chappell, Susannah Thorpe, Nick Hawes and Aaron Sloman (Eds.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

Grounding Symbols: Learning and Grounding 2D Shapes using Cell Assemblies that emerge from fLIF Neurons

Fawad Jamshed[1] and Christian Huyck[1]

Abstract. Symbols are the essence of a language and need to be grounded for effective information passing including interaction and perception. A language is grounded in a multimodal sensory motor experience with the physical world. The system described in this paper acts in a simple virtual world. The system uses multimodal interaction by using vision and language, to interact with the outer world. Two simulations are given in which the system learns to categorise five basic shapes. In the second simulation, the system labels these categories, grounding them. Both simulations performed work perfectly on all tests.

1 Introduction

Although many aspects of human cognition and language are still a mystery, guidance can be taken from human processing where possible [1]. Infants ground their language by interaction with the physical world, associating speech patterns with objects and actions in the world [2]. An infant initially learns categories through observing the world. Humans use a very complex communication system. It is widely accepted, that the human communication system is symbolic, learnt, compositional and multimodal. Understanding the underlying principles of the human communication system requires an understanding of the mechanisms with which the words' meanings are rooted in reality.

The system, whether it is human or artificial, must learn to categorize input into bins. So, when an agent senses an item, it must categorise it as, for instance, a dog. This allows the system to generalize its behaviour so that it can react similarly to different instances of that category. One explanation of this process of category learning is provided by Hebb [3]. A cell assembly, a reverberating neural circuit, represents the category or concept. The cell assembly is learned by modifying synaptic weights using a Hebbian learning rule, which states the synaptic weight between two co-firing neurons is increased. The categories are formed in response to environmental stimuli. Thus, the reverberating circuits of cell assemblies are formed by groups of features that tend to travel around together. That is, the cell assembly for dog is formed by a group of features that dogs tend to have. When a dog is presented to the system, the dog cell assembly becomes active. This cell assembly formation is a form of sensory motor toil [4]. New categories are learned formed through real time, feedback-corrected, trial and error; the agent uses senses and motion to develop those categories.

These concepts are entirely subsymbolic. They are formed in direct response to the environment, and can be formed by animals that do not process symbols. However, humans do process symbols, and these symbols can be linked to the sub symbolic cell assemblies. This process of labelling is essential for symbol grounding.

A great deal of artificial intelligence concentrates on symbol processing, but no simple symbol processing system can be truly intelligent [5]. Instead, a host of implicit knowledge is needed, and that knowledge is stored in the underlying cell assemblies, and relations between cell assemblies. Thus, a truly intelligent artificial system must have grounded symbols [6]. Once a system has some grounded symbols, it can then use symbols to learn new symbols and new categories [7]. This is the benefit of language, a person can learn from language without actually having to experience the environment. An agent can learn where someone lives by being told their address. This symbolic theft, based on the categories that have been derived via sensory motor toil, makes the system much more powerful. According to symbolic theft, new symbols are formed by combining already grounded symbols [4]. Some basic categories must still be learned by sensory motor toil, and then by using symbolic theft, new symbols can be acquired.

Symbols are grounded by multimodal interaction with the world [8]. Multimodal communication can involve vision and language integration to ground mental concepts in the physical world [9]. This multimodal integration, between language and visual representation enables humans to acquire and use words in context and makes communication possible by establishing coherence between mental states and the physical world. Thus not only visual items need to be learned, they need to be labelled to ground them.

In this paper, a system that learns to categorise visual objects by sensory toil is presented. The system then labels these categories, providing the basic structure of symbol grounding. This system is implemented entirely in simulated neurons (see section 2.1). Cell assemblies (see section 2.2) emerge from the system's interaction with the environment. The learning of these visual items is described and evaluated in section 3. These items are then labelled; this process is described and evaluated in section 4. Conclusions are then drawn and future work discussed.

[1] School Of Engineering and Information Sciences- Middlesex University, The Burroughs –London, UK.
Email: {f.jamshed, c.huyck}@mdx.ac.uk.

2 Theoretical Background and Previous Work

The system is based on a simple, but realistic, neural model. This model is the fatiguing Leaky Integrate and Fire neuron. In response to environmental stimuli, these simulated neurons can form cell assemblies. The precise form of the response is governed by particular Hebbian learning rules.

2.1 Fatigue Leaky Integrate and Fire Neurons

The fatiguing Leaky Integrate and Fire (fLIF) neuron is a relatively simple, though accurate, model of biological neurons. The fLIF neuron is an extension of the more widely used LIF model [10]. This is, in turn, an extension of the integrate and fire model [11].

The model used in this paper makes use of discrete cycles. A cursory description is provided and a fuller explanation can be found elsewhere [12]. Each neuron has some activation, which it receives from other neurons. If a neuron has enough activation at the end of a cycle, it will fire, spread activation to connected neurons, and lose all its energy. Neurons are connected to other neurons with unidirectional, weighted connections. A firing neuron passes the activation of the weight of the connection. If a neuron's activation is less than the threshold, it will not fire but some the retained activation will leak away for the next cycle. This is modelled by calculating the activation as describe in equation 1.

$$A(t) = A(t-1)/D + C. \qquad \text{(Equation 1)}$$

Where A(t) is the activation at time t. It is the sum of the activation at time t-1, reduced by a decay constant D, and C, the sum of incoming activation of all active neurons that are connected to a given neuron and fired at time t-1. The value of C is determined by multiplying the incoming activation from other firing neurons at time t-1 with the associated weights of those links.

fLIF neurons also fatigue just like biological neurons. If a neuron fires regularly it becomes harder to fire. This is modelled by increasing the threshold of a neuron when a neuron fires, then reducing it when the neuron does not fire.

2.2 Cell Assemblies

A Cell Assembly (CA) is the neural basis of a symbol [3]. It is a subset of neurons that have high mutual synaptic strength enabling neurons in the assembly to persistently fire after external stimulation ceases. In the simulations discussed in this paper, a small subset of all the neurons represents a symbol. If many of the neurons in it are firing, the symbol is active.

Cell assemblies give a sound answer to the neural representation of two types of memory, long-term memory and short-term (or working) memory. The firing of many neurons in an assembly is the neural implementation of short-term memory; this high frequency and persistent firing makes the cell assembly active.

2.3 Hebbian Learning

The basic learning mechanism used in these experiments is the compensatory learning rule that has been described elsewhere [13]. The overall rule is derived from a correlatory rule that combines Hebbian and anti-Hebbian mechanisms. That is, the synaptic weight between two neurons increases when the neurons co-fire, and decreases when the pre-synaptic neuron fires and the post-synaptic neuron does not. Over time, the synaptic weight reflects the likelihood that the post-synaptic neuron fires when the pre-synaptic neuron does. This correlatory mechanism uses a modifier that forces that combined synaptic weight leaving a neuron toward a constant. This is done by making the increases smaller and the decreases larger when the total synaptic weight of a neuron is greater than constant; and making the increases larger and the decreases smaller when the total synaptic weight is less than constant. This rule makes synaptic weight a resource.

2.4 CABot

The simulations described in section 3 and section 4 are an extension of the first version of the Cell Assembly Robot (CABot1) [14]. The main aim of CABot is to develop an agent in simulated neurons, which can take natural language as input and interact with the environment without any external help. By interacting with the environment, it is hoped that it can learn the semantics of the environment sufficiently well to improve language processing.

For CABot1, a virtual 3D environment was established based on the Crystal Space games engine. Two agents were placed in the environment, the first controlled by a user, and the second was the CABot1 agent. All processing in CABot1 was done by a complex network of fLIF neurons, though it emitted symbolic commands to the Crystal Space stub.

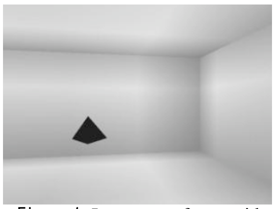


Figure1. Instances of pyramid

A complete description of CABot1 is beyond the scope of this paper but further information can be found elsewhere [15]. A total of 21 sub-networks are used to subdivide the tasks of vision, natural language parsing, planning, action and system control.

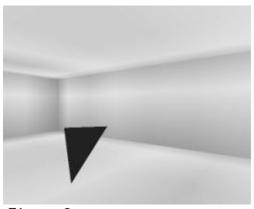


Figure2. Instances of stalactite

Proceedings of the International Symposium on AI-Inspired Biology, Jackie Chappell, Susannah Thorpe, Nick Hawes and Aaron Sloman (Eds.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

The important subnets for the purposes of this paper are the vision nets and the word nets. There are three vision subnets, a simulated retina, a primary visual (V1) cortex and a secondary visual cortex (V2). These systems were hard coded, so there was no learning. Visual input was in the form of a bitmap representation of a view of the game from the agent's perspective. In particular, the secondary visual cortex subnet was set to recognise *pyramids* and *stalactites.* If one of these was present in the game, a CA in V2 ignited. There were several position and size dependent CAs associated with both *pyramid* and *stalactite*. Figure 1 and figure 2 shows instance of *pyramid* and *stalactite* respectively.

Similarly, the parsing component had CAs for words. In the game, the user issues natural language commands to tell the agent what to do. There was a noun instance subnet to store semantic roles during parsing.

3 Learning Visual Items

Learning was introduced in to the system with the help of six visual sub nets. The five shapes used are: pyramid, stalactite, diamond, square and right angle triangle. Currently the vision system consists of six nets which are Input net, Retina net, V1 net, V1A net, V2 net and V4 net. Each of these six sub nets performs a unique function, takes input only from the prior net (the environment for Input), and passes input to the subsequent net.

The Input net displays the input from the environment whereas the Retina net is a series of OnOff and OffOn detector; these nets are unmodified from CABot1. The V1 net is position dependent and detects first order features of a solid shape in the picture. These features included including oriented edges and angles, and several new angle detectors were added from CABot1. The V1A net is a position independent model of the V1 net, merely showing the presence of the feature without location. The V2 net detects the second order features, a three way combination of edges and angles. For example, it detects a horizontal edge, a right angle, and a vertical edge are all simultaneously present. There were 13 such second order detectors. The V4 net identifies the shape of an object with the help of the second order features which are detected in the V2 net. The detailed working of the vision system is defined below.

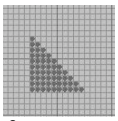


Figure 3a. Right Triangle

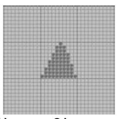


Figure 3b. Pyramid

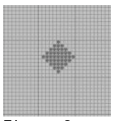


Figure 3c. Diamond

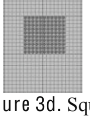


Figure 3d. Square

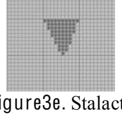


Figure3e. Stalactite

Figure 3a, 3b, 3c, 3d, and 3e shows the instances of right angle triangle, pyramid, diamond, square and stalactite respectively.

The Input net gets the input from the system in the shape of bits and displays it on the screen. The input is usually in the shape of pictures but shapes can be hard coded in the system. The Retina net is a biological plausible model of the OnOff and OffOn detectors that are found in biological systems; it gets the input from input net, and feeds its output to the V1 net. Three different types of OnOff and Off On detectors are used in the Retina net 3 by 3, 6 by 6 and 9 by 9 detectors.

V1 is position dependent and gets the input from the OnOff detectors and identifies the first order features e.g. edges and angles of a solid shape in the picture. The V1 net responds to different types of edges and angles presented. The connections from the V1 net were made position independent by introducing the V1A net and making random connections from each V1 CA to the corresponding V1A CA. The V1A net has direct connections from the V1 net only. The neurons of the V1A net have a low decay rate of 1.01 to promote even a small firing of neurons in V1.

The V1 net and the retina net used are modified versions of the V1net and the retina net of CABot1. More CAs are introduce in the V1 net including vertical edge CAs ,and four types of right angle CA. Whereas theV2 net, the V4 net and the VT net (described below) were introduced for this experiment.

The V2 net gets input from the V1A net, which is the position independent version of the V1 Net. When a three or four edged shape is presented, each CA of the V2 net gets three inputs from three CAs of the V1A net. CAs of the V2 net are only ignited when all of three CAs in the V1A net is active when a shape is presented. The V2 net output is used as an input to the V4 net where final shape is determined.

The V4 net is the final part of the vision system where all the shapes are discriminated. The V4 net and the V2 net are fully connected which means each of the CAs in V2 net is connected with all CAs of V4 and vice versa. Learning is carried out between V2 and V4 nets.

The same topology was used within V1A net, V2 net and V4 net. In this topology, twenty percent of the neurons used were inhibitory neurons while eighty percent of the neurons used were excitatory. In the case of the V4 net, there are inhibitorier intra-CA connections to promote winner takes all situations so that only one CA will eventually ignite.

In order to prevent a CA from winning on two different shapes, a slow learning approach is used. Due to the low learning rate, the weights do not grow rapidly. Hence the V3 net CAs does not commit to the V4 net rapidly, but over a longer period, reducing the likelihood that they will form incorrect associations.

3.1 Experiment

During the simulations, all five shaped are learned. Instances of those shapes are presented to as visual input. The input neurons are clamped on for 250 cycles. Activation passes through the retina, V1, V1A and V2 subnets. The activation then passes to the V4 net; as the weights from V2 to V4 are initially low, no V4 CA ignites initially. However, each of these nets slowly accumulates activation, and eventually one ignites, and persists. As it inhibits the other CAs, it is the only one that persists. The winner takes all strategy prevents one shape being committed to more than one CA of theV4 net. During this phase of persistent firing, the weights from V2 and back to V2 are raised via Hebbian learning.

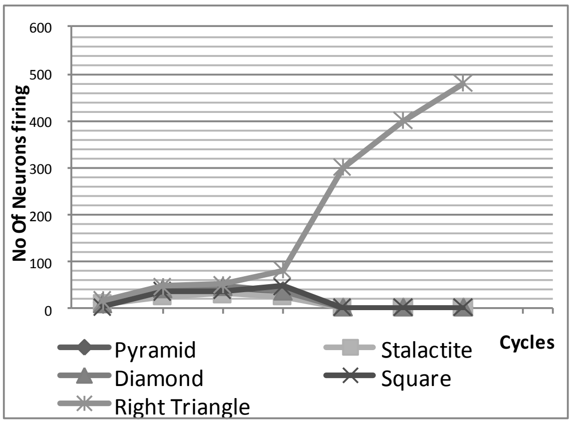
An instance of each of the five shapes is presented to train the network. It is then tested by presentation of other instances of the shapes. Table 1 shows the result of a successful test when an instance of visual pyramid and of visual diamond are presented; the number of neurons firing in the corresponding CAs of V4 show that specific CAs are committed to pyramid and diamond respectively, whereas the other CAs of the V4 net do not respond. The correct ignition of the CAs in the V4 net shows the system has learned these shapes.

Table 1. No of neurons firing in the V4 net during testing phase

Neurons Firing	Pyramid	Stalactite	Square	Diamond	Triangle
When pyramid is presented	490	0	0	0	0
When Diamond is presented	0	0	0	479	0

The simulation is termed successful when all of the five CAs of the V4 net are committed to the five different distinctive shapes, whereas the quality of the success is determined by how well these five shapes get learned CAs to respond when different shapes are presented.

Graph 1. Example of winner takes all in this case its triangle



Graph 1 shows a typical example of winner takes all strategy (in this case its triangle). Initially all the CA compete with each other but slowly one CA gets promoted which inhibits other CA neurons making them more harder to fire.

The training runs for 1250 cycles where each of the five shapes was presented for 250 cycles. The system is reset at 250 cycle intervals with all neural activation and fatigue reset to 0. The connections between the V2 net CAs and the corresponding V4 net CA are adjusted and learned using Hebbian learning. The learning is bi-directional with weights on connections from the V2 net to the V4 net and weights from the V4 net to V2 net being learned at the same time.

The test runs for 2500 steps. During the testing part of the simulation, shapes were presented in a random order. After presenting the complete set of five shapes in 1250 cycles, another set of shapes was presented, randomly again, for another 1250 cycles.

3.2 Results

The test is fully automated and runs without human intervention. The test was conducted 28 times. The result is calculated using the correct numbers of corresponding CAs of the V4 net being fired when a shape is presented to the system.

The success rate among three shapes diamond, square and rectangle is one hundred percent. Each of the shapes i.e. diamond, square and rectangle, gets committed to a different CA each time they are presented and during testing each of the committed CAs responds correctly to different shapes presented each time. Due to the very similar features, pyramid and stalactite shapes do get committed wrongly sometimes and the CA which is being committed to one of the shapes presented first also sometimes responds to the second shape presented afterwards. The success rate among the shapes of pyramid and stalactite is 75 percent.

4 Labelling Items

Labelling is a simple but crucial part of the symbol grounding problem and associates the correct labels with semantic categories [16]. This simulation shows how learned semantic categories for shapes get appropriate labels attached to them. These associations are then tested. The use of semantic CAs and label CAs is consistent with linguistic research where a word has a semantic pole and a phonological pole [17].

4.1 Experiment

The system is presented with different pictures of instances of shapes and after detecting the appropriate first and second order features, the correct shapes are learned (as described in section 3). These learned shapes are then associated with a label by presenting an instance of shapes and its label simultaneously. The connections are adjusted and learned using Hebbian learning.

Learning is bi directional so the label can be used to retrieve the shape, and the shape can be used to retrieve the label. Each shape and label is represented by a CA of 500 neurons in the V4net and the label net respectively. Every 5th neuron in each of shape and label CAs has 3 random connections with the CAs of each label in the label net and with the CAs of each shape in the V4net. The values of other parameters used in the experiment are as shown in the table 2.

Table 2. Values of the parameters used in the experiment

Parameter	Value
Decay rate	1.45
Learning rate	0.2
Fatigue Rate	1.2
Fatigue Recovery rate	1.4
Activation threshold	4.2

The correlatory learning rule is used to learn the connections. According to the correlatory learning rule, when pre-synaptic and post-synaptic neurons fire, the weight is increased, and when the pre-synaptic neuron fires and the post-synaptic neuron does not fire the weight is decreased. Initial weights of connections between the shapes and the labels are set low (0.01) so the weight can be adjusted to the appropriate level.

During testing, the semantic input to the system is turned off to check if the right semantic CA is ignited when a label is presented. Similarly to test the connections from the semantics to labels, input to the label net is turned off to see if the correct label ignites.

During learning, each instance of shape and label was presented for 250 cycles. Each epoch of learning and testing for all the shapes and the labels lasted for 1250 cycles. After 4 epochs of learning the system is tested for 4 epochs. In the first 2 epochs of testing, the semantic input to the system is turned off to test the label to semantic connections. In the next 2 epochs of testing, the semantic input to the system is turned on but the input to the label net is turned off to test semantic to label connections.

Table 3a. Visual pyramid is presented when Input to the label net is off.

	Pyramid	Stalactite	Square	Diamond	Triangle	Label Pyramid	Label Stalactite	Label square	Label Diamond	Label Triangle
No Of Neurons Firing	480	0	0	0	0	459	0	0	0	0

Table 3b. Label stalactite is presented when Input to the V4 net is cut off.

	Pyramid	Stalactite	Square	Diamond	Triangle	Label Pyramid	Label Stalactite	Label square	Label Diamond	Label Triangle
No Of Neurons Firing	0	465	0	0	0	0	472	0	0	0

Table 3a shows the result of a run when the input to the Label net is switched off and an instance of visual pyramid is presented to the system. When an input is presented in V2, a CA in the V4 net is ignited, which, in turn, ignites the correct associated Label CA even though it has no external input. Table 3b shows the results of a run when the visual input to the system is turned off and input is applied to the Label net. The CAs of the Label net also ignites the corresponding CAs of the V4 net even though there is no visual input to the V4 net. These tables clearly show the appropriate connections are being learned.

4.2 Results

The results were calculated by counting the number of neurons fired at 249th cycle of presenting a shape or label. During learning and testing different instances of the shapes were used. On all 10 tests that were carried out, all the right CAs ignited when the system is presented with the corresponding CAs. This correct ignition of the CAs shows the correct association between shape and label CAs is learned appropriately.

5 Conclusions and Future Work

As has been pointed out, to some extent the Symbol Grounding Problem has already been solved [18]. That is, artificial systems have learned categories and labelled them. Of course, the problem still remains unsolved in that no artificial system currently grounds symbols nearly as effectively as a typical human does.

This paper describes an early step toward symbol grounding. Like people, this system processes using neurons. Also like people, this system learns new visual categories. Finally, like people, this system labels those categories.

As compared to other models (for example [19], [20], [21], [22]), our model uses biologically plausible neurons, and the system is more generic in nature (as it is not only bound to spoken words). The model described in this paper has the flexibility to be used with incremental learning and can use environmental feedback to create or readjust what is already learned [23].

This paper has shown progress on the symbol grounding problem. The symbol grounding problem is clearly a problem that humans must solve, but little work in the study of natural cognition centres on this issue. This is probably because this is so effortless for humans. Humans learn new categories and instances constantly throughout life. They do this implicitly and explicitly; those memories may be labelled or not; and the labelling is also usually effortless. However, as greater understanding of natural cognition evolves, these complexities will be better understood.

The promising results of these experiment shows that the technique and model used for these experiments can be used effectively to solve other aspects of the symbol grounding problem. Other aspects of that need to be addressed in order to solve the SGP are functional symbol grounding, symbolic theft, hooking, alignment and environmental feedback. Functional symbol grounding involves grounding a symbol in a given context. Symbolic theft involves grounding new categories from splitting or merging already grounded symbols. For example by combining stripes with horse, a new category, zebra, can be created and vice versa. Hooking is when a category can have more than one representation; for example a chair looks very different from the front and back. An intelligent agent should be able to ground all representations such that object is categorised regardless of the angle of input. Hooking also refers to synonymy [24] with multiple labels applying to a single object or category. Alignment involves coping with an unfamiliar symbol by using the information from already grounded symbols. For example if an agent has not seen an object before, using its already grounded symbols, the agent can infer the meaning of the new symbol. Environmental feedback means an agent should be able to learn or readjust already grounded symbols.

Though, this list is not exhaustive, addressing these issues will yield an agent that can much more effectively ground symbols. Such an agent will be able to make its own decisions without little help from the outside world.

References

[1] Clark. A. (1997). “Being There: Putting Brain, Body, and World Together Again” Cambridge, MA: MIT press.

[2] Bloom. P. (2000). “How Children Learn the meaning of Words” Cambridge, MA: MIT Press.

[3] Hebb, D. (1949). The Organization of Behavior. New York: Wiley.

[4] Cangelosi, A., Greco, A., & Harnad, a. S. (2000). “From Robotic Toil to Symbolic Theft: Grounding Transfer from Entry-Level to Higher-Level Categories” Connection Science , 143-162.

[5] Searle, J. (1980). Minds, Brains and Programs. Behavioral and Brain Sciences , 417-424.

[6] Harnad, S. (1990). “The Symbol Grounding Problem”. Physica D , 335-346.

[7] Massé, A. B., Chicoisne, G., Gargouri, Y., Harnad, S., Picard, O., and Marcotte, O. (2008). “How is meaning grounded in dictionary definitions?” In Proceedings of the 3rd Textgraphs Workshop on Graph-Based Algorithms For Natural Language Processing). ACL Workshops. Association for Computational Linguistics, Morristown, NJ, 17-24.

[8] Dillenbourg, P., Traum, D., and Schneider, D. (1996). “Grounding in Multi-modal Task-Oriented Collaboration” In proceedings of the Euro AI and Education Conference.

[9] Pastra, K. (2004). “Viewing Vision-Language Integration as a Double-Grounding Case” In Proceedings of the American Association of Artificial Intelligence (AAAI) Fall Symposium on “Achieving Human-Level Intelligence through Integrated Systems and Research”, pp. 62-67, Washington D.C. USA.

[10] Amit, D. (1989). “Modelling Brain Function: the world of attractor networks”. Cambridge University Press.

[11] McCullouch, W., & Pitts, W. (1943). “A Logical Calculus of Ideas Immanent in Nervous Activity”. Bulletin of Mathematical Biophysics , 115-133.

[12] Huyck, C. (2007). “Creating Hierarchical Categories Using Cell Assemblies.” Connection Science , 1-24.

[13] Huyck, C. (2004). “Overlapping Cell Assemblies from Correlators”. Neurocomputing , 453-459.

[14] Huyck, C. (2008). “CABot1: a Videogame Agent Implemented in fLIF Neurons” In proceedings of 2008 7th IEEE international Conference on Cybernetic Intelligent Systems, (pp. 115-120). London.

[15] Huyck, C., & Byrne, E. (2009). “CABot1: Technical Report”. Middlesex University.

[16] Jamshed, F. and Huyck, C. (2009). “Grounding Symbols: Labelling and Resolving Pronoun Resolution with fLIF Neurons” International Conference on Machine Learning and Applications.

[17] Langacker, R. (1987). Foundations of Cognitive Grammar Vol. 1. Stanford, CA: Stanford University Press.

[18] Steels, L. (2007). “The Symbol Grounding Problem has been solved. So what's next.” In M. deVega, Symbols and Embodiment: Debates on Meaning and Cognition. Oxford University Press.

[19] Roy, D. (2003). “Grounded Spoken Language Acquisition: Experiments in Word Learning”. IEEE Transactions on Multimedia, 5(2): 197-209.

[20] Roy, D. and Reiter, E. (2005). “Connecting language to the world” Artif. Intell. Vol. 167. 1—12.

[21] Yu, C. and Ballard, D. (2004). "A Multimodal Learning Interface for Grounding Spoken Language in Sensory Perceptions", ACM Transactions on Applied Perception, 1, 57-80.

[22] Yu, C. and Ballard, D. H. (2004). “On the integration of grounding language and learning objects” In Proceedings of the 19th National Conference on Artifical intelligence. AAAI Press / The MIT Press, 488-493.

[23] Belavkin, R. and Huyck, C. (2009). “A Model of Probability Matching in a Two-Choice Task Based on Stochastic Control of Learning in Neural Cell-Assemblies” In Proceedings of the Ninth International Conference on Cognitive Modelling.

[24] Ostler. N, Atkins.B. (1991). "Predictable Meaning Shift: Some Linguistic Properties of Lexical Implication Rules" Proceedings of the First SIGLEX Workshop on Lexical Semantics and Knowledge Representation, Springer-Verlag.