

Intrinsic Representation: Bootstrapping Symbols From Experience

by

Stephen David Larson

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2003

© Massachusetts Institute of Technology 2003. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
August 22, 2003

Certified by
Patrick H. Winston
Ford Professor of Artificial Intelligence and Computer Science
Thesis Supervisor

Accepted by
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Intrinsic Representation: Bootstrapping Symbols From Experience

by

Stephen David Larson

Submitted to the Department of Electrical Engineering and Computer Science
on August 22, 2003, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The vision of intrinsic representation is to produce systems that can learn the meanings of things without explicit instruction. Such systems, building representations up from low-level perceptual information, can discover rules and relationships that exist in their environments autonomously. In order to accomplish this goal, a new perspective must be taken on how to design knowledge representations that are capable both of semantic completeness and knowledge generation. A model of intrinsic representation is proposed as an adaptive knowledge representation scheme that tackles these issues. An implementation using self-organizing maps serves as a proof of concept for this model.

Thesis Supervisor: Patrick H. Winston

Title: Ford Professor of Artificial Intelligence and Computer Science

Acknowledgments

Contents

1	Introduction	13
1.1	Steps Towards a Solution	14
1.2	Overview	15
2	The Symbol Grounding Problem	17
2.1	Semantic Completeness	20
2.2	Knowledge Generation	21
3	Related Work	23
4	Model of Intrinsic Representation	25
4.1	Overview of Model	25
4.2	Theoretical Foundation	26
4.2.1	Regularity in the Environment	26
4.2.2	Information Spaces	28
4.3	Key Points	29
5	Tools for Implementation	31
5.1	Self Organizing Maps	31
5.1.1	Connection To The Problem	31
5.1.2	The Basic Algorithm	32
5.2	Hierarchical Growing Self Organizing Maps	33
5.2.1	Connection to the Problem	34
5.2.2	Map Growth	34

5.3	Clustering	37
5.4	Blocks World	37
5.4.1	Why Choose the Blocks World?	39
6	Architecture and Implementation	41
6.1	Architecture	41
6.1.1	Blocks World to Self-Organizing Map	41
6.1.2	Self-Organizing Map to Cluster Set	43
6.1.3	Cluster Set to Cluster Association	44
6.1.4	Modality A to Modality B	44
7	Discussion	47
7.1	Symbol Grounding and Intrinsic Representation	47
7.1.1	Semantic Completeness	48
7.1.2	Knowledge Generation	48
8	Contributions	49
A	Multidisciplinary Background	51
A.1	Philosophy: Interactivism	51
A.2	Computational Neuroscience	53
A.3	Systems Neuroscience	55
A.4	Machine Learning	57

List of Figures

4-1	Model of Intrinsic Representation. At the bottom, information from the environment impinges on the most peripheral sensory neurons, creating signals in an information space. Two information spaces, i and j , are shown side by side. Over time, the regularity in the information space is stored on a map through a process of self-organization. From this map, clusters of high similarity can be segmented and reified as their own units. Associations are created and strengthened based on the co-activation between clusters in different information spaces, creating networks of activation. As members of these networks, clusters can be used as inputs in new information spaces and can be associated with other co-occurring units in different systems, allowing reactivation of local clusters from afar.	27
5-1	A self-organizing model set. An input message X is broadcast to a set of models M_i , of which M_c best matches X . All models that lie in the vicinity of M_c (larger circle) improve their matching with X . Note that M_c differs from one message to another. (from fig.1 in [12]) . . .	32
5-2	Insertion of Units (from [7])	34
5-3	Hierarchical Growth (from [7])	35
5-4	A cluster tree produced by UPGMA. Data points A-F are clustered. .	37
5-5	A simple 2D Blocks World.	38

6-1	Blocks World to Self-Organizing Map. Data from the blocks world is read out into a self-organizing map in the form of real-valued normalized vectors.	42
6-2	Self-Organizing Map to Cluster Set. A map is divided up into clusters A, B, and C using a clustering algorithm that operates on its units. .	43
6-3	Cluster Set to Cluster Association. Clusters in different maps are associated together.	44
6-4	Example of a trained representation. Representation has been trained with eye fixated on arm and no blocks. Arm and eye were moved together throughout the space. Clusters were formed, and further arm-eye movement was used to create associations between the clusters. Figure shows how the arm can then attend to a location the eye is looking at by utilizing the trained representation.	45
A-1	The Perception-Action cycle. (from [9], modified without permission)	56

Preface

I believe that the the major hurdles in the path of achieving human-level artificial intelligence are conceptual rather than technological. I share the growing concern among some researchers that the pathway to achieving such a goal is still, after fifty years of AI research, hazy at best.

As a consequence of this, the hope of this thesis is to shed some theoretical light on the obstacles along that pathway.

Chapter 1

Introduction

“Once a problem is described using an appropriate representation, the problem is almost solved.” [21]

How the brain represents the building blocks of thought is one of the remaining unsolved mysteries of human existence. That the brain creates representations of some kind is clear; we observe many organisms acting on prior information. Such behavior necessitates that organisms appropriately store and retrieve information. In humans, these representations are well suited to enable the performance of an endless number of tasks. What is less clear is how the brain represents this information in such an advantageous way.

We want to build intelligent systems that know what things are. Before we can do this we must have a better idea of what it *means* to know what things are. In particular, we need to know what it means for a brain to know what things are.

Modern representations of the world around us have been greatly expanded by the ability to encode information in a computer. Because computer models are dynamic and can be updated quickly on demand, sophisticated computer models have been made possible. Today, computers are used for modeling a variety of complicated things— car engines, economic theories, and the human genome are just a few. What we have discovered is that within narrow, well-defined problem domains, finding the right representations is a matter of reductionism. Within these domains, good representations are found by first understanding the basic principles of the desired

application and then matching an appropriate encoding scheme. However, in problem domains where the basic principles are unclear, finding good representations is much more difficult. How the brain builds models about things in the world that enable it to carry out tasks for survival is currently one of those uncharted domains. Unfortunately our ability to make some powerful models has not yet translated to an ability to make the kind of powerful models the brain can.

1.1 Steps Towards a Solution

This thesis presents a model of knowledge representation that is concerned with what things are. The model gathers the knowledge not by being explicitly told by a programmer, nor by making rule-based inferences with a formal system. Rather, it gathers knowledge from low-level sensory experience with an environment. It does this in an unsupervised way. By taking advantage of the statistical regularity present in environments, the model discovers symbols. These symbols then associate together to form networks of activation that tend to correspond roughly to an object in the environment. Once these networks have been established, their patterns of activation can be detected and learned by higher level systems, enabling new knowledge to be built on top of existing knowledge.

This model serves as a substrate for the “middle layer” in a system concerned with human-level intelligence. Networks of activation correspond to symbols and are discovered in a bottom-up fashion. Those networks can then be re-activated either from the bottom up based on what cluster an input falls into, or from the top down based on what associations a cluster has made. Such a system therefore exhibits the property that it allows recall of an object to occur using some of the same machinery that is used when observing the object. There is much evidence from neuroscience that this is a property shared by the brain. Further work may show that reasoning about an objects can be explained as the re-activation of these networks in light of other re-activated networks, the combinations of which allow novel conclusions to be reached and appropriate motor behaviors to be selected.

In addition to describing this model in more detail, this thesis describes a working testbed implementation of the model. Experiments that are carried out on the testbed serve as an existence proof that such a model can work in simulation.

1.2 Overview

The preceding preview should serve as a helpful guide to the rest of my thesis. In chapter 2, more specifics about the limitations of modern representations are given. The symbol grounding problem is used as a motivating problem, and the absence of semantic completeness and knowledge generation are identified as the issues of greatest concern. Related work is discussed in chapter 3. We turn to a more detailed explanation of the model of intrinsic representation in chapter 4. In chapter 5, the significant components of the implementation are described. Chapter 6 ties the elements in chapter 5 together and explains the architecture and implementation of the model. In chapter 7 the results are analyzed, and in chapter 8 the contributions made by this thesis are summarized.

Chapter 2

The Symbol Grounding Problem

One of the drawbacks of traditional symbolic AI reasoning systems is their domain specificity. What causes this? Harnad, with his description of the symbol grounding problem, sheds some light on this question. He describes the problem as the following:

“How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads?” [10]

Harnad is saying that a symbol is merely a label for a larger body of knowledge that is difficult to describe without that label. The label should at no point be confused for the larger body of knowledge to which it refers.

An example of the problem he describes is seen by considering a person reading the word “chair”. This action summons a mental image of the object. However, the perception of the printed word by itself is not equivalent to the perception of the mental image. One can think at length about a chair beyond the five letters of its English representation. One can consider its various aspects, chairs that one is familiar with, and so on. Those abilities must be fueled by information that is obviously not contained in the letters c-h-a-i-r. Because of this extensive ability to recall information about a chair, it is clear that there is more to our mental representations of words than their printed image.

For a symbolic system, however, the printed image is almost exclusively what represents an object. To date, symbolic systems lack the ability to intrinsically interpret the meaning of a data object representing a chair the way humans can. At best,

a symbolic system may represent a data object with reference links to other data objects in order to model its different properties. For example, a chair may be the member of a category of ‘furniture’ and may have a ‘wooden’ texture. However, these associated properties of a chair are merely other symbols—other labels that humans give meaning to. This limitation does not prevent us from teaching a limited-domain systems to function with competency at specific tasks such as playing chess, because we are able to define all the relevant information about the symbols being manipulated. The legal moves of a chess piece are easy to program into a computer, and do not ever vary. However, when we attempt to design systems to operate outside of a limited domain, we find that its functionality is limited by both by the information that can be given to it, as well as the way that information can be stored.

The symbol grounding problem challenges us to distinguish between a symbol as analogous to markings on a piece of paper and the mental processes that allow us to consider its meaning. It also forces us to consider that we know more about the symbols we mentally manipulate in our world than we are able to describe.

For example, when we look at a toy block, we have a wealth of information about it at our fingertips. We can produce language concerning the block—we can say its name, its color, its shape, and describe its purpose. We also can solve problems with it, such as figuring out if it would fit into a particular box, or if it could be used to hold up other blocks. We know what it feels like to pick it up, and we know what happens if we balance it on its corner.

As you read the previous paragraph, you recall these different aspects of a block for yourself. By invoking the symbol “block”, you have keyed into your personal knowledge about a physical object. Is the knowledge representation in your brain equivalent to the description given above? Are there really English sentences stored in your brain that *tell* you what a block feels like in words? Or is it represented in some other way that later and separately enables a description in language? Is it possible that your sense of what a block feels like is captured in a way more similar to the actual sensation than to any particular words?

As a solution to the symbol grounding problem, Harnad proposes to under-gird

the computer descriptions of symbols with non-symbolic representations derived from sensory experience. Such non-symbolic representations are real-valued, and are derived from the information received from our most peripheral neurons. The benefit of using sensory experience as the lowest common denominator representation is that the meaning of symbols can be defined as the recollection of the experiences associated with them. For example, a toy block can be represented as the combination of all of the sensory interactions one has had with a chair.

Clues from the field of neuroscience suggest that some form of this kind of representation is found in the brain. “Recent functional brain imaging studies suggest that object concepts may be represented, in part, by distributed networks of discrete cortical regions that parallel the organization of sensory and motor systems.” [14] We might imagine that a symbol, such as the printed word form of an object, acts as an index that activates appropriate combinations of cortical brain systems. The “meaning” of a printed word symbol would come from the process of these distributed cortical systems contributing information about aspects of the object. With the ability to represent symbols as combinations of sensory experiences, we may also achieve a deeper understanding of how the human brain can integrate information from different sensory modalities.

For AI research, the symbol grounding problem is an obstacle to reaching systems that are capable of more human-like knowledge representations. Unfortunately, we still do not have a clear idea of what qualities we should aim to give symbols in order to make them more grounded. Below I suggest that the qualities of 1) semantic completeness, and 2) knowledge generation are good candidates to begin with. If these issues are resolved, the symbol grounding problem will play a much less significant role in the development of AI systems. These two issues will be defined and discussed in turn. Before defining these terms, however, let us first examine what we mean when we talk about representation.

Representation

One of the goals of this thesis is to explore new ways of representing knowledge. In order to do this, it is useful to have a more precise notion of what it means to represent knowledge.

Davis says that one of the important roles of a knowledge representation is as a surrogate or a stand in for something else [6]. In addition, a knowledge representation is a model of something else. I will frequently use the term knowledge representation not only as something a computer uses to represent knowledge but also as something that the brain uses represents knowledge.

How do symbols relate to knowledge representation, and what do we mean by symbols exactly? In my usage, symbols are a particular way of representing things. Symbols are a representation that can be isolated as individual units. Some knowledge representations use symbols to create a kind of language that is useful for capturing knowledge. Such systems are what are referred to as symbol systems or traditional symbolic systems.

2.1 Semantic Completeness

Semantic breadth is defined as the ability for a representation to capture knowledge about a broad range of topics. It can be used as a comparative property of representations—representations can be more or less semantically broad than one another. Consider an example of abstraction in concepts. The concepts wooden-thing, block, and rectilinear-block are ordered from more general to more specific, and from more abstract to less abstract. We would call wooden-thing more semantically broad than rectilinear-block, because the former captures knowledge about a broader range of topics. Blocks are not the only wooden-things, so are trees and chairs.

The corollary to semantic breadth is semantic depth. Semantic depth is defined as the ability for a representation to capture knowledge at a fine resolution. It can also be a comparative property. With the wooden-thing example, rectilinear-block is more semantically deep than wooden-thing because it gives us more detail about the

type and shape of the wooden-thing.

In the wooden-thing example, semantic breadth and semantic depth are inversely related. What we would like is to achieve representations that are the best of both worlds— both broad and deep. Such idealized representations we will refer to as *semantically complete*. Our target for semantically complete representations will be those that must be in brain that allow us to know about both a broad range of topics and at the same time be able to understand a great deal of detail about each of them.

The symbol grounding problem points out why symbol systems are less semantically complete than brains— because as discussed above, symbols usually mean more to brains than they do to computers. Symbolic systems have limited semantic breadth because they are trapped into narrow domains. They have limited semantic depth because we are unable to give computers as much detailed information about things as we are able to collect about them ourselves.

2.2 Knowledge Generation

While semantic completeness is a concern regarding modern knowledge representations, knowledge generation is a concern regarding the systems that operate on them. The concern is this: while some AI systems are capable of generating previously unknown information, we have not yet achieved a system that gives us back significantly more information than we put in. Human beings can do this. Human beings have the ability to generate knowledge on top of knowledge in a cycle that appears limitless. It appears as if the ratio of information out to information in is very high. The question is, how can we create systems that share this capability?

Modern AI's weakness is solving problems where the symbols are unavailable or outside of the existing "symbol realm". The problem of knowledge generation suggests that in order to make a system more general, we must create a system that can "fill in the gaps" present in the representations it presently has. A desirable system along these lines would be able to identify information that it lacks, and do something to acquire, represent, and utilize that knowledge effectively.

These representations might be formulated in such a way that by themselves, without being operated on, they are so difficult to describe, that they are in fact their own best description. [13] If this is possible, how else will we be able to capture these representations, but with a system that puts them into place for us? Thus having such a system of knowledge generation may be more than just a convenience, it may be a necessary component of a system that hopes to approximate human intelligence.

Chapter 3

Related Work

Other work towards establishing meaning for symbols has been conducted. Agre and Chapman [1] investigated the instantiation of general symbols using indexical-functional aspects. Aspects were intended to provide a meaning for objects in a world relative to their usefulness for an agent. Objects were instantiated as different aspects depending on the way in which the agent could use them. This work was a significant step towards breaking free of a symbolic representation rooted only in meaningless symbols. Indexical-functional aspects added extra meaning into the system. Symbols could now be tracked with relation to the goals of the agent, which simplified the process of planning. Agre and Chapman's work falls short of creating the kind of representation envisioned here, however, because 1) their functional representations are still rooted in the domain of playing the game, and 2) their representations are encoded in syntactic structures rather than real values.

Gary Drescher's work [8] looked at how a system can build knowledge on top of knowledge through interaction in a simple micro-world with a hand, an eye, and objects. Much of what Drescher had to say is relevant, and in many ways this thesis is an attempt to improve on his results. The main limitation of the system he invented was the separation between his knowledge representation and his means of interacting with the world. His schemas were created in a theoretical space that did not impose much constraint on how data from the hand related to data from the eye. As a result, too many of his primitives were equivalent, and could fit into several places in his

schemas. Because of this and his departure from a more faithful model of sensation, Drescher also experiences some effects of the symbol grounding problem that we seek to avoid.

Coradeschi [5] addressed the problem of symbol anchoring, a similar problem to that of symbol grounding, but done on autonomous robot systems. An emphasis was placed on the use of multiple sub-systems interacting to produce correspondences between a symbol and sensory percepts. While the focus of this work is on the process of object recognition, we are interested more by the form and utility of representations made from the interaction of a system with the world.

De Beule [2] approached the symbol grounding problem directly and built a blocks world in order to approach the problem. The system is given information about the objects such as their positions and their constituent properties. The system returns syntactic structures describing notable objects in a world such as “the red square moving to the right”. While this produces interesting correlations, it is still a representation limited both by the amount of information given to it by design and the amount of generalizable knowledge that it exports. We are seeking a system that will draw conclusions on the basis of its own interaction with the world, not with pre-packaged information such as position.

Roy [19] has shown ways that a system can learn to correspond images and words. While this is most along the lines of the goals of the present work, Roys system has some limitations that we seek to avoid. Roy does not pursue interactive aspects of the learning process. His system does not inquire about the content of the world in order to resolve conflicts in its understanding. Roy seeks to produce correspondences rather than a model. While this pursuit will be satisfied developing relevant correspondences between observations and symbolic representations, the greater goal is to understand how such representations can be self-generated by a system through exploration.

Chapter 4

Model of Intrinsic Representation

The model of intrinsic representation is a system designed to create symbols from the regularity inherent in the world. Its key differences from traditional symbolic systems are 1) symbols are discovered from the statistical processing of experience, 2) symbols are equivalent to statistical regularities found in information spaces, and 3) symbols carry their context with them by being situated in information spaces.

4.1 Overview of Model

Figure 4-1 is the key diagram for the understanding of the model of intrinsic representation. At the bottom of the diagram, sensory arrays receive streams of data from the outside world. Such arrays could be imagined as a patch of skin or light sensitive cells in the retina. As data comes into the system through these sensors, it travels to a subsystem devoted to organizing and storing the regularities in the data. This subsystem arranges these regularities with respect to their similarity, placing highly similar regularities near each other, and dissimilar regularities farther apart. After a critical period, clusters of high similarity group the regularities into sets. These clusters are considered the symbols of this system. At this point, data in the incoming stream is treated as a trigger for the activation of the cluster of which it is a member. As clusters are activated by incoming data, they are associated together by their frequency of coincidence. The more often two clusters are active simultaneously, the

more associated they will become. The resulting trained system thus treats incoming patterns as members of a class, and can react to the activation of that class.

4.2 Theoretical Foundation

This model is built on top of a few theories about the way information works in brains. First is the theory that there is enough statistical regularity in the environment from which brains can reasonably form useful symbols. Second is the notion of an information space as a definable entity in either brains or computers. Let us cover each theory in turn.

4.2.1 Regularity in the Environment

The information brains collect about their environment contains statistically detectable patterns. There are several reasons to believe that this claim is true.

From a systems perspective, it seems that the brain must be doing something to reduce the informational entropy that it must deal with coming in from its primary sensory areas. Psychologists have commented that one of the amazing things about the brain is not that it collects so much information, but rather that it manages to filter out all the irrelevant information that it is constantly awash in. If we could not accomplish this we would never be able to focus on anything, and we would lose the survival game. Thus it seems like in order to make sense of the world, the brain must be selective about the information that it allows to flow through its synapses.

But the brain does have some help. While the amount of information coming from the world is immense, it could be much worse. The fact is that there is a significant amount of regularity in environments that brains have evolved to deal with. We do not live in a world of white noise. We live in a world where matter has permanence, and tends to be localized. Laws of physics are consistent. Thus there are certain kinds of information out of all possible information that are prevented by our environments from ever reaching us. The information that I have stepped on a rock and accidentally fallen *upwards* is an example of something that evolving brains

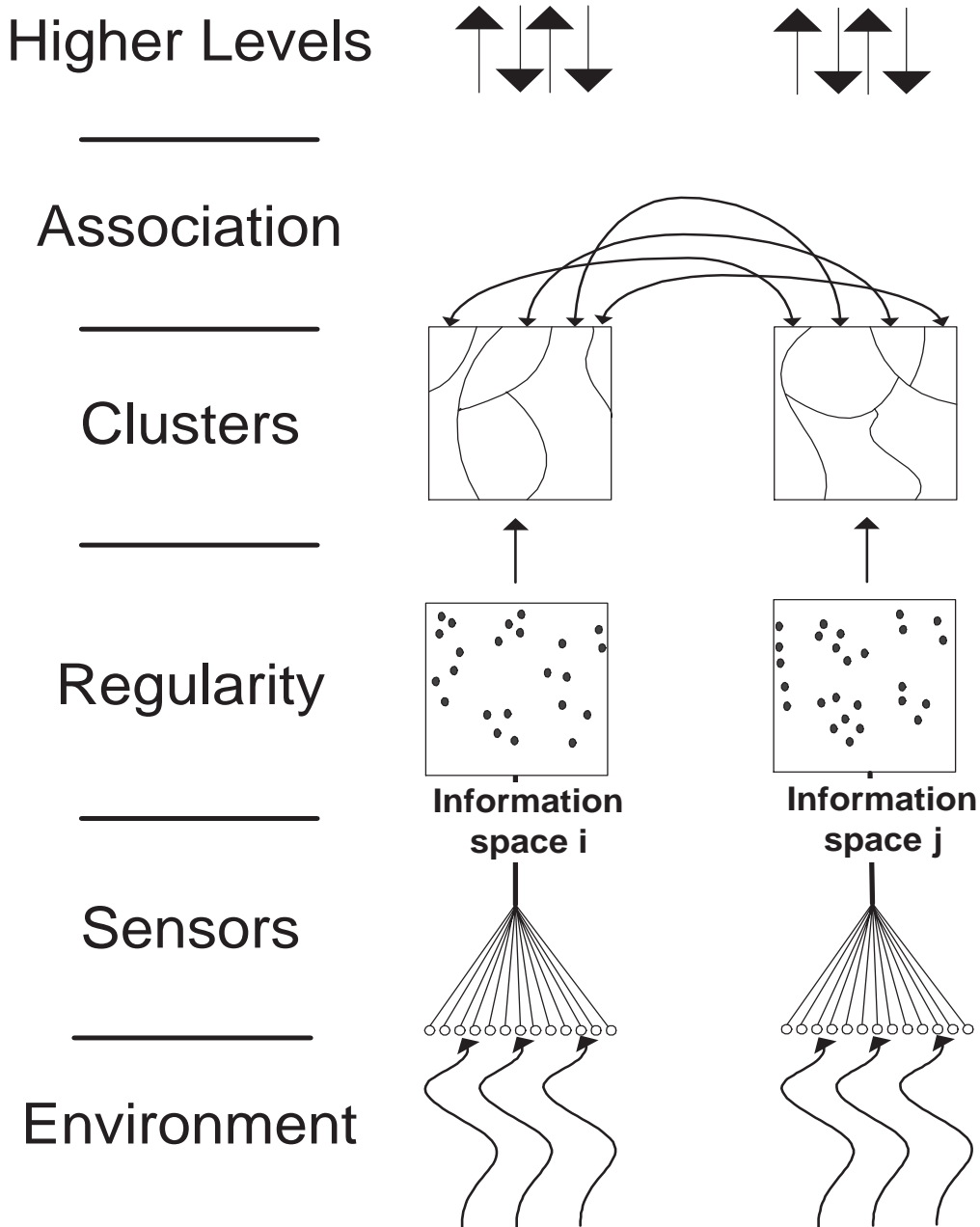


Figure 4-1: Model of Intrinsic Representation. At the bottom, information from the environment impinges on the most peripheral sensory neurons, creating signals in an information space. Two information spaces, i and j, are shown side by side. Over time, the regularity in the information space is stored on a map through a process of self-organization. From this map, clusters of high similarity can be segmented and reified as their own units. Associations are created and strengthened based on the co-activation between clusters in different information spaces, creating networks of activation. As members of these networks, clusters can be used as inputs in new information spaces and can be associated with other co-occurring units in different systems, allowing reactivation of local clusters from afar.

have not had to compensate for. The regularity in our environments gives us a leg up in trying to make sense of the world than in an hypothetical world where purely random information impinged on our sensory cells.

4.2.2 Information Spaces

I introduce the concept of an information space as a useful way of compartmentalizing streams of information in brains. The best physical metaphor for an information space would be the data traveling along a bundle of wires. A snapshot of the space would be an n -dimensional vector capturing the values of each wire at a given instant in time. Information spaces can be thought of as vector spaces applied to the brain, and all of the language of Linear Algebra is applicable in analyzing them.

For example, let us think about the information flowing from the optic nerve into V1. If we were to take a cross section of the optic nerve just before it enters V1, we could analyze the dynamics of all the synapses that compose the optic nerve at that point. If we imagine each synapse as a wire carrying a voltage, we can imagine that each synapse will at any given time, have some real-valued level of excitation in some range. You could plot each level of excitation on an axis, or you could consider all the wires at once as a point or a vector in an n -dimensional space, where n is equal to the number of wires. As data flows through this cross section, we can imagine that this point or vector travels through some kind of trajectory through this n -dimensional space. All possible points in this n -dimensional space comprise the information space.

Information spaces can be defined arbitrarily, so long as you can select a group of wires. You can define an information space as the synapses leaving V1 entering V2. You can define an information space as the synapses inside a particular part of V1 at a particular place. The trick is identifying the spaces that are useful.

The choice of information spaces defines the kinds of knowledge that a given instantiation of this model will learn about. The information space leading into V1 is very different than the information space leading into the primary sensory areas. Different things will be salient to different spaces.

It may seem odd that while this model claims to be a knowledge representation,

it is undefined what kind of knowledge it will be representing. This is part of the paradigmatic shift of intrinsic representation. We can gather knowledge about whatever and however many spaces we choose to include in our system.

4.3 Key Points

As described above, the key differences between the model of intrinsic representation and traditional symbol systems are:

1. Symbols are discovered from the statistical processing of experience
2. Symbols are equivalent to statistical regularities found in information spaces, and
3. Symbols carry their context with them by being situated in information spaces.

Let us deal with each in turn:

Symbols are discovered from the statistical processing of experience. As mentioned in section 2, symbols are units of information that stand in as a surrogate for some other information. In almost every traditional symbol system, the symbols must be provided before-hand. In the model of intrinsic representation, clusters of regularity are symbols. Because these clusters are discovered by the system in an unsupervised way, the symbols do not have to be provided beforehand.

Symbols are equivalent to statistical regularities found in information spaces. The nature of the symbols in traditional symbol systems are as tags whose interrelationships are provided by a human designer. Symbols which are formed from statistical regularities come with their interrelationships already determined by their informational nature. Symbols will represent certain things in the world not because a designer finds them useful, but because those things are the most statistically salient in the given information space.

Symbols carry their context with them by being situated in information spaces. Traditional symbol systems require context as an extra parameter to make sense of a symbol. Symbols may mean different things in different contexts. With intrinsic representation, however, because symbols are derived from information spaces, they are inextricably linked to those information spaces, and carry no individual meaning outside of them. As a result, you cannot have a symbol without its context. Furthermore, this context is useful. Symbols can be compared for similarity by examining their relative locations in their information space.

Chapter 5

Tools for Implementation

This chapter points towards chapter 6 by separately describing the significant algorithms and technologies used in the implementation.

5.1 Self Organizing Maps

The Self-Organizing Map (SOM) algorithm performs unsupervised learning on a set of incoming n-dimensional input vectors. In its basic form, it is visualized as a sheet-like two-dimensional array of cells. As the SOM is trained, the cells become tuned to patterns in the input, and the resulting map reflects an organization that can be thought of as a 2D projection of the most salient relationships in n-dimensional space.

5.1.1 Connection To The Problem

How are SOMs related to the subject of symbol grounding? The biggest potential for SOMs in AI are as adaptive knowledge representations. Because they change fluidly based on their input, they are robust against noise.

The broad vision for where SOMs fit into a field of AI that has foundered after discovering that neither top-down nor bottom-up approaches work by themselves is right in the middle. SOMs are a plausible choice for an interface between symbolic systems and connectionist systems. How to go about doing this is the subject of this

thesis.

Self Organizing Maps stem from a line of research into statistical learning algorithms that include such methods as Vector Quantization and Principal Components Analysis. Its innovation has been its multidisciplinary influences. Self Organizing Maps try to mimic some of the functionality reported in the sensory feature maps of the cortex. While it is sometimes called a neural networks algorithm, its units do not function in a manner that are consistent with standard models of neurons. SOMs are designed to be a biologically plausible system that is built at a level of abstraction above that of the neuron level. The idea of modeling the activity of a group of neurons all at once in a high-level, easy to implement algorithm is one of the reasons that SOMs are so popular. [11] has a complete survey of the applications of the SOMs idea that have been created in the relatively short life-span of the algorithm.

5.1.2 The Basic Algorithm

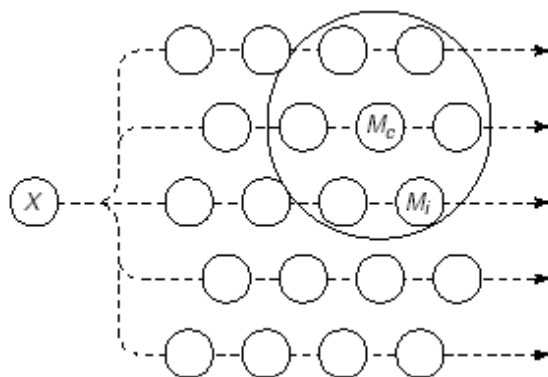


Figure 5-1: A self-organizing model set. An input message X is broadcast to a set of models M_i , of which M_c best matches X . All models that lie in the vicinity of M_c (larger circle) improve their matching with X . Note that M_c differs from one message to another. (from fig.1 in [12])

Here is how the basic algorithm works. Each training iteration begins with the selection of an input pattern, $x(t) \in \mathfrak{R}^n$, where t is a discrete-time coordinate. Each cell i in the map has a model vector, $m_i(t) \in \mathfrak{R}^n$ associated with it. After initializing all cells in the map to a random distribution:

1. Find the winner cell $c(t)$ whose model vector, $M_c(t)$, best matches $x(t)$.

$$c(t) = \operatorname{argmin}_i \|x(t) - m_i(t)\| \quad (5.1)$$

In the basic model, Euclidean distance is used as a similarity measure.

2. Discover neighborhood $N_c(t)$, which are the cells surrounding the winner cell c
3. For each $i \in N_c(t)$, update the model vector as follows:

$$m_i(t+1) = m_i(t) + \alpha(t)[x(t) - m_i(t)] \quad (5.2)$$

4. Otherwise, the model vector stays the same:

$$m_i(t+1) = m_i(t) \quad (5.3)$$

This process repeats for all the patterns available.

$\alpha(t) \in [0, 1]$ refers to the learning rate at time t . Both α and $N_c(t)$ are typically reduced as the learning process evolves to allow for more focused representations towards the end of the training session.

5.2 Hierarchical Growing Self Organizing Maps

Hierarchical Growing Self Organizing Maps are an extension to the basic SOM idea that allows maps to increase the number of cells over time and to spawn new maps. It is described in detail by its creators in [7]. Let us first discuss the extra machinery that allows maps to grow, and finally discuss the machinery that allows new maps to be spawned.

Whereas in the basic SOM model, maps have a set number of cells that does not change throughout the training phase, HGSOMs begin with a 2x2 map which grows over the course of training.

5.2.1 Connection to the Problem

The extension of the SOM idea in this way introduces the idea that an information space can have a hierarchical similarity structure. Additionally, the notion of the quantization error is a nice objective measure for the modification of a map based on data, free from human bias. Lastly, the choice of a fixed topology with the basic SOM is a design decision that is difficult to justify. Improving the SOM algorithm by having that decision made by the statistical structure of the data makes one less thing to worry about.

HGSOMs were designed as an extension of the SOM algorithm with specific applications in mind. The creators of the HGSOM wanted to discover more about the high-dimensional spaces that they were trying to analyze with SOMs. One of their goals was to examine the similarity of a corpus of Time Magazine articles, and to discover what kind of relationships between them the algorithm would discover. They were particularly interested in the unsupervised formation of categories of articles. The HGSOM turned out to be a useful tool to aid them in this endeavor. [7]

5.2.2 Map Growth

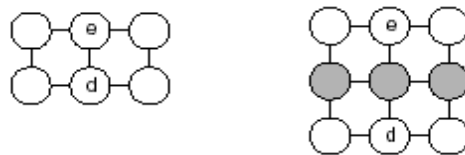


Figure 5-2: Insertion of Units (from [7])

The basic SOM algorithm is carried out on the map to begin. Every λ iterations, a growth phase is started. During the growth phase, an error cell, e , is selected. Next, the most dissimilar neighboring cell, d is selected. Finally, a new row or column of cells is created between e and its most dissimilar neighbor d .

The selection of e involves the computation of a value known as the quantization

error (qe) for each cell. Here is the formula for qe:

$$qe_i = \sum_{x_j} \| m_i - x_j \| \quad (5.4)$$

For each cell i , we compute the similarity between every input vector x_j that is mapped onto cell i and compute the sum of the difference between the model vector m_i and x_j .

The error unit e is the cell with the highest qe on the map during the growth phase.

$$d = \operatorname{argmax}_i (\| m_e - m_i \|), m_i \in N_e \quad (5.5)$$

The dissimilar neighbor d is found by finding the cell that is the most different of all the neighbors of the error cell, N_e .

The new row or column is filled in with model vectors that are the average of the model vectors on either side.

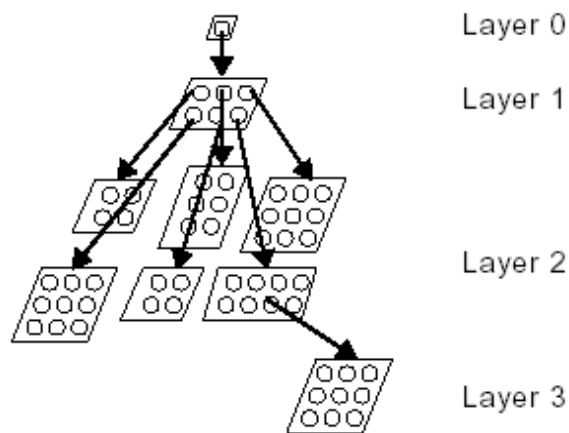


Figure 5-3: Hierarchical Growth (from [7])

Hierarchical Growth The process of deciding when to spawn new child maps and what units to spawn them from also involves the quantization error. When a map

reaches a stopping criterion, it spawns a child map from each cell that represents a set of too diverse input vectors. New maps are trained on the input vectors that are mapped to their parent.

The stopping criterion for a map is a function of its mean quantization error (MQE).

$$MQE_m = \frac{1}{n_U} \sum_{i \in U} qe_i, n_U = |U| \quad (5.6)$$

The MQE of a map is the mean of all units' quantization errors for the subset U of the maps' cells onto which data is mapped.

The stopping criteria used to decide when a map is ready to start spawning child maps is:

$$MQE_m < \tau_1 \cdot qe_u \quad (5.7)$$

where qe_u is the qe of the cell u in the upper layer. For the first-layer map, u is treated as a single cell through which all input into the first-layer map flows, and its quantization error, qe_0 keeps track of how diverse the data the map has received is.

The criteria used to decide which cells spawn new maps is:

$$qe_i < \tau_2 \cdot qe_0; \quad (5.8)$$

Two parameters, τ_1 and τ_2 serve as controls on the depth/shalowness of the resulting HGSOM and granularity of the data representation, respectively.

5.3 Clustering

Clustering is a general term for a large class of unsupervised learning techniques. The goal of clustering is to discover groups of data points or “clusters” that are similar to each other.

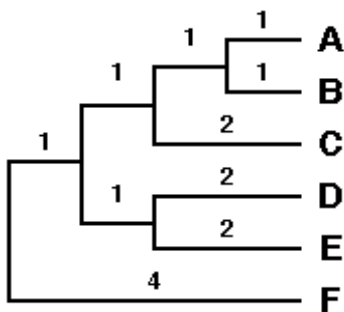


Figure 5-4: A cluster tree produced by UPGMA. Data points A-F are clustered.

The Unweighted Pair Group Method with Arithmetic Mean (UPGMA) style of clustering is one of the most popular and simple methods used.[18] Its output is a tree, as shown in fig. 5-4. Each branching point in the tree corresponds to a cluster. The further to the left in the figure, the fewer clusters you have, and the more data points each cluster includes. The full tree for a data set is sometimes thought of as its “similarity structure”.

5.4 Blocks World

A Blocks World is, at its most basic, a problem domain. Typically it includes blocks of various colors and shapes, capable of being placed in different locations in a space. Blocks are capable of being stacked upon one another if they are properly shaped. Typically a robot arm is used to manipulate the blocks.

The most well known implementations of an AI system using a blocks world as a problem domain was SHRDLU, a program developed by Terry Winograd at MIT. [20] The program’s focus was on language interaction with a system that could do limited reasoning about a block’s world. The system was able to carry on what appeared to

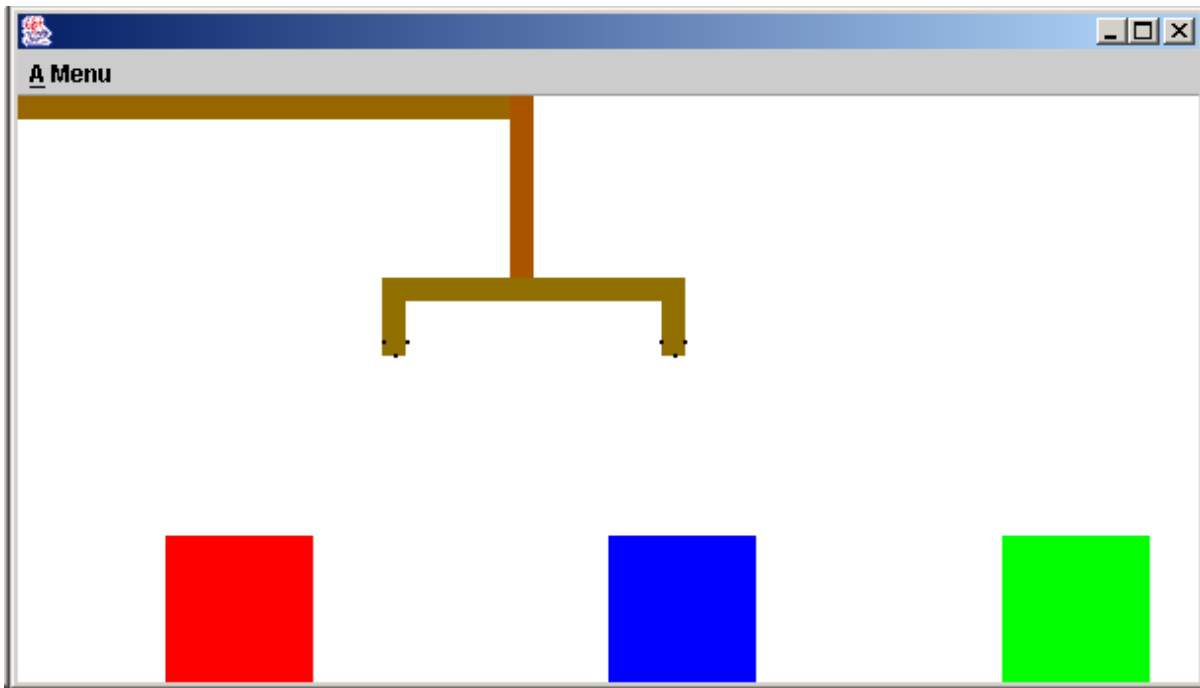


Figure 5-5: A simple 2D Blocks World.

be a decent dialog with a person about the blocks world– it could answer questions and carry out commands.

While Winograd’s world did prevent cubes from being stacked on top of pyramids, it did not attempt to model the physical reality of the problem set much more faithfully. The arm could pick up any block that did not have something else resting on it, without concern for grip. The arm never placed a block in a position where it could be affected by gravity, thus gravity was not simulated. The arm was essentially prevented from making any “mistakes” by the reasoner, and this simplified the model greatly.

Since then, various symbolic AI systems have used problem spaces similar to a blocks world to conduct experiments in AI.[4] A blocks world is a good choice for a purely symbolic system. It is easy to model the state of the world using well-defined symbols.

More recently, the notion of a blocks world is being taken more literally, and experiments are being conducted using blocks worlds that more faithfully model Newtonian Dynamics. [2]

5.4.1 Why Choose the Blocks World?

There are several reasons that the blocks world makes sense to use as the problem domain for a project concerned with symbol grounding. The blocks world as it was originally used is, pun not intended, symbolic of the limitations of symbol-only systems. Traditional AI systems built to work within the blocks world provide good examples of the symbol grounding problem. While these systems were successful to an extent, their major limitation was their lack of representational generality, and their inability to generate knowledge. Consequently, systems like SHRDLU could not scale up to incorporate more interesting worlds.

Once Newtonian Dynamics are added to the blocks world, the domain provides a reasonably good analogue to those that infants are presented with. We could imagine a system using this domain as an infant at a particular level of development and use faithfulness to an infants true behavior as a criteria for success. It is unlike game domains such as chess or expert domains such as medical diagnosis which are only comparable to well-developed brains. Operating from the bias that we need to understand how simple skills are formed before we can understand how complicated skills work, a domain that simple minds can understand is desirable.

When you consider what it would take to make a robot that plays with blocks, you begin to realize that even this seemingly simple domain has enough variability to be an interesting problem space. Marvin Minsky has used the idea of a blocks world extensively in *The Society of Mind* [15]. In his theories, a child at play with blocks learns problem solving skills that can be used to solve different problems later in life. In order to build towers out of blocks, children have to have intuitive understandings of simple physics, how to make wholes from parts, etc. Thus there is hope that the more light we can shed on the things computers need to know to play with blocks, the more we may learn about how to make them reason about more complicated domains.

The substance of what we will learn is likely to be more about representation than it will be about reasoning. Modern success in AI appears to have solved most

of the reasoning problems that involve simple representations. The real progress to be made in the next fifty years of AI research will be representational—to try and find the representations that allow us to approach brain-like behavior while remaining manageable enough that engineers can reasonably build systems using them.

Chapter 6

Architecture and Implementation

This chapter discusses first the architecture and then the implementation of a computer program demonstrating the model of Intrinsic Representation.

6.1 Architecture

The architecture of the program is structured into four major areas:

1. A Blocks World Simulation
2. A Self Organizing Map
3. A Cluster Set
4. Cluster Associations

Each of these blocks feeds into the other in progression. Below, connections between these areas as well as the areas themselves are discussed.

6.1.1 Blocks World to Self-Organizing Map

Data flows into the system from the environment. A simple 2D blocks world provides the environment for the present system. In the blocks world, there is an arm with a simple grip useful for grasping at objects, much like the arcade novelty that allows

players to try and catch prizes by controlling a robot arm. There is also an eye, whose focus is represented by a square that can be moved around the environment. The eye and the arm are the sources of sensory and motor interaction with the world. There are also blocks in the world able to be picked up and stacked on top of one another. Simple physics modeling is in place to enable such constraints as gravity.

Both the eye and the arm are equipped with sensors. Every sensor is normalized, and thus reads a real value between zero and one. The eye has retina sensors arranged in a 2D array that register the red-green-blue value of any spot they are over. The eye also has sensors that tell it its horizontal and vertical orientation, mimicking the information the brain receives from the muscles that move the eyes.

The arm has proprioceptive sensors and tactile sensors on its grip. The proprioceptive sensors tell it how far it is extended both horizontally and vertically, mimicking feedback from muscles and joints. The tactile sensors tell if an object is colliding with the sensor or not.

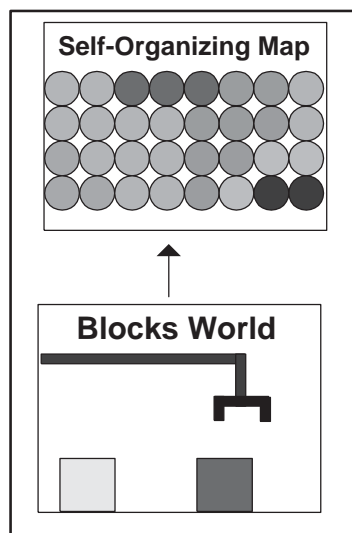


Figure 6-1: Blocks World to Self-Organizing Map. Data from the blocks world is read out into a self-organizing map in the form of real-valued normalized vectors.

A vector of sensor values is read out from the eye and arm each time their values change. The vector has n dimensions, where n is the number of independent sensors on the device. As mentioned, this vector is normalized. Thus, its relevant information can be thought of geometrically as its “angle” in an n -dimensional space.

As vectors are read out from a given device, they are received by a self-organizing map. The map is specialized to its input only in the respect that its dimensionality must match. As vectors are received, the map's self-organizing process iterates in the way described in section 5.1.2.

6.1.2 Self-Organizing Map to Cluster Set

The map is allowed to self-organize for a significant amount of time. Once an indication that the map is representing its information space at an acceptable level, the map is switched to read only mode. Using the model vectors in the cells of the 2D array, a clustering algorithm is used to separate the major clusters in the space.

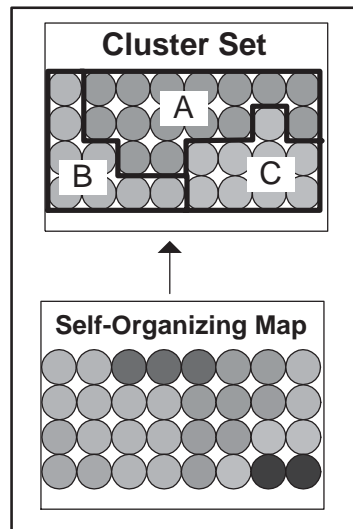


Figure 6-2: Self-Organizing Map to Cluster Set. A map is divided up into clusters A, B, and C using a clustering algorithm that operates on its units.

Each of the clusters on the map is given its own unique identifier, and a new data object is created to represent it. This data object keeps track of the similarity measure of the cells inside the cluster, as well as the indices of the cells it contains. Clusters are stored together in Cluster Sets, which also keep track of the relationships between clusters.

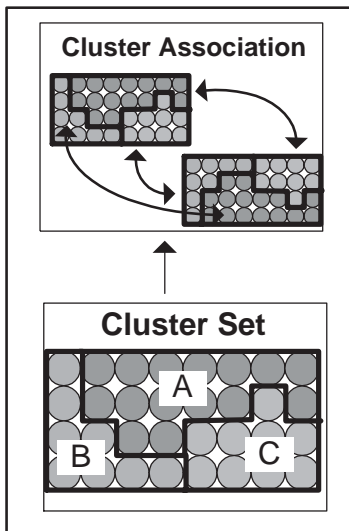


Figure 6-3: Cluster Set to Cluster Association. Clusters in different maps are associated together.

6.1.3 Cluster Set to Cluster Association

Once clusters have been stored for a map, they can be associated with clusters in other maps. Clusters are never associated with clusters from the same map. The association process is extremely simple. When clusters are active at the same time, a counter labeled with the names of those clusters is incremented. Clusters are active when input matches most closely with a cell within the cluster. The cluster association system can identify the clusters most associated with an individual cluster.

6.1.4 Modality A to Modality B

Once the representation has been trained, it can be used to send signals between modalities and exhibit behavior that neither modality separately would be able to accomplish as easily. Figure 6-4 illustrates how this is possible for the simple example of getting the arm to move to the position the eye is currently looking at.

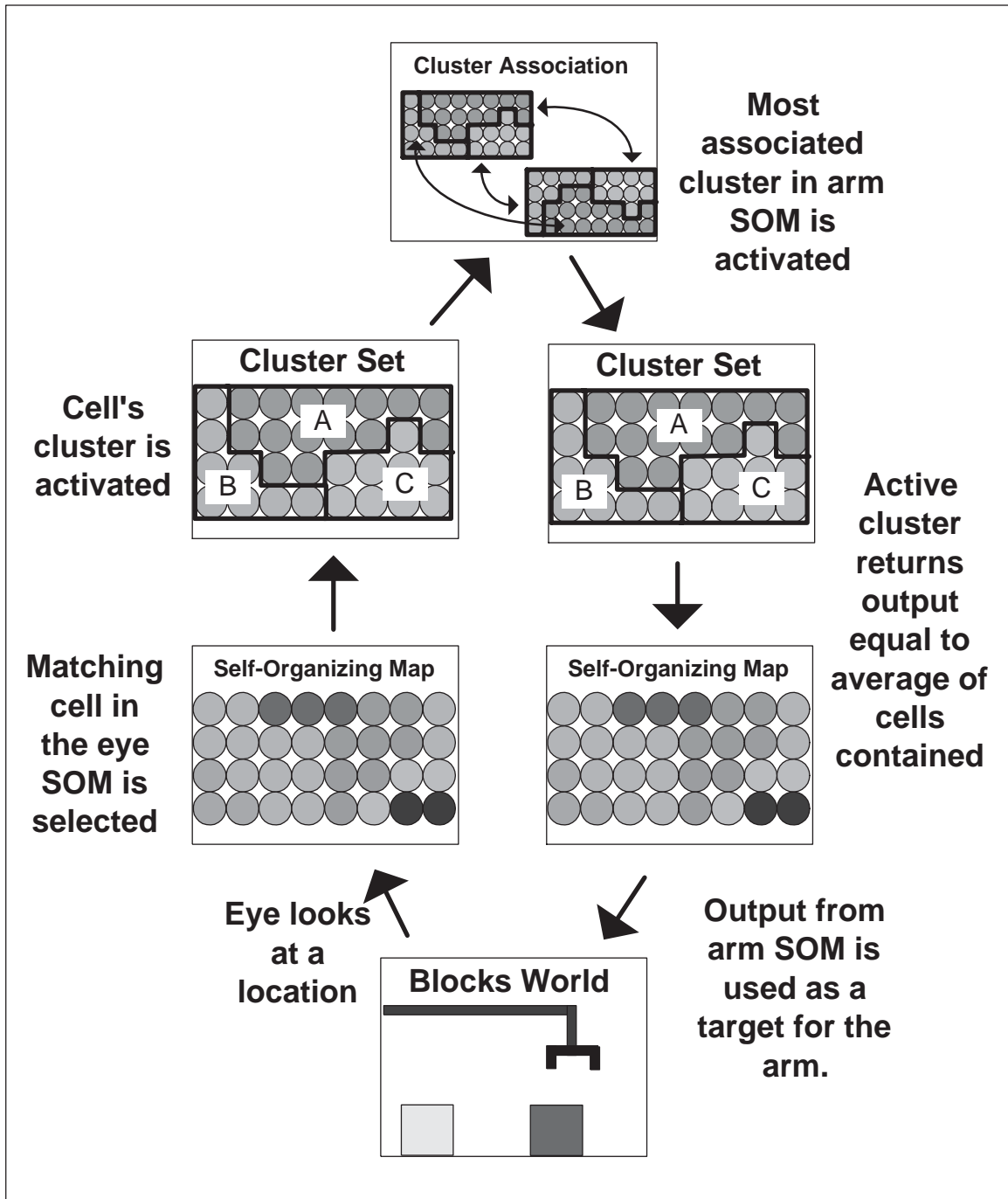


Figure 6-4: Example of a trained representation. Representation has been trained with eye fixated on arm and no blocks. Arm and eye were moved together throughout the space. Clusters were formed, and further arm-eye movement was used to create associations between the clusters. Figure shows how the arm can then attend to a location the eye is looking at by utilizing the trained representation.

Chapter 7

Discussion

Now that the architecture and implementation of the model of intrinsic representation has been presented, let us go back and examine this system in light of the issues raised in chapter 2.

7.1 Symbol Grounding and Intrinsic Representation

What progress does this model make towards solving the symbol grounding problem? Does the representational system bring us closer towards the criteria of semantic completeness and knowledge generation? I would argue yes on all counts.

As discussed at length in chapter 2, the symbol grounding problem complains that symbols are generally disconnected from the kind of meaning that people impose on them. The model of intrinsic representation does not have this flaw. Our symbols—the clusters formed from regularity in the environment—are directly connected to the experience of the system. In fact, the symbols cannot exist without experience as they are created through a process of experiencing the world. Additionally, the relationship between the symbols and the information spaces they reside in is significant. The symbol cannot carry its meaning outside of the information space that it is created. The information space provides the context in which that symbol is meaningful. This

is desirable because symbols in this model can never be separated from their contexts.

7.1.1 Semantic Completeness

As we recall from section 2.1, semantic completeness is the goal of creating representations that are both semantically broad and semantically deep. Intrinsic representation is certainly semantically broad— it can learn symbols in any information space it is pointed towards. How about semantic depth? As mentioned in section 4.2.2, how information spaces are structured to feed into one another as the learning process in the model involves will determine how much it will learn. Thus, the semantic depth is the product of structuring information spaces correctly, though initial signs indicate that the potential semantic depth is significant.

7.1.2 Knowledge Generation

In section 2.2, we described the property of knowledge generation as the ability for a representation to generate more information than is given to it. Intrinsic representation is better at knowledge generation than traditional symbolic systems. Consider the setup of the model: choose the proper input streams for the most efficient learning. The rest is left up to the statistical analysis of the input streams and a few objective criteria concerning when different phases should begin and end. The design decisions are few. This fact, combined with the fact that there is no need for a human to encode each piece of data that the system will learn with is a major win.

Chapter 8

Contributions

In this thesis, I have:

- identified some limitations of modern representations that prevent them from being more general,
- built a model that addresses those limitations and enables more general representations to be formed autonomously, and
- implemented a test bed for my model that serves as an existence proof that the model can be instantiated.

Appendix A

Multidisciplinary Background

A.1 Philosophy: Interactivism

A radical school of thought called Interactivism [3] seeks to overthrow popular conceptions of mental representations as simply scribbles on a tablet. Its position is essentially a more pointed and aggressive version of the symbol grounding problem. The core ideas as relevant to this topic are as follows:

- A symbol is useless as bits that sit in memory.
- A symbol is useful as a part of a larger system that it interacts with.
- Therefore, in order to talk about a symbol's meaning, we have to talk about the larger system that it resides in.

This can be understood by making an contrast with the standard Turing Machine model of computation. A Turing Machine is a simple computation machine that reads ones and zeros off of a tape that is fed to it. In such a model, we can take two positions on the value of the ones and zeros on the tape when they are separated from a machine that can interpret them. We can either say that the data on the tape has intrinsic representational value separate from the Turing machine, or that the data on the tape has no intrinsic representational value separate from the Turing machine. The first position would argue that information is stored on the tape that can be accessed

at any time, and thus the tape has the ability to represent information. The second position would argue that the data on the tape has no intrinsic representational value because without the Turing machine, the data cannot be accessed—its just a lump of ones and zeros— and thus does not represent anything *intrinsically*. The question then boils down to what one thinks it means to have intrinsic representational value.

Interactivism takes the position that the tape does not have intrinsic representational value, and the only intrinsic representational value comes from the combination of the tape and the machine—the action of reading of the tape and acting on its instructions. It goes further to say that mental representations are likely to be of the sort that have intrinsic representational value.

It is not necessary to accept all tenets of Interactivism to appreciate the assumptions it is attempting to challenge. Data on a tape is only useful at the moment it is being processed. Information in memory is only useful while it is in the processor. Such is the nature of a serial processing device like the modern computer. Brains, on the other, hand, with information stored along synapses¹, have not only the ability to process information in parallel, a commonly attributed feature, but because of the nature of their representations, also have the ability to allow access to the data in those representations to occur simultaneously. A cluster of CPUs operating in parallel still have to read data from memory a chunk at a time, suggesting that the number of chunks of data in use any given clock cycle is equal to the number of processors. But the nature of storing information along the links of a network suggests that the number of “chunks” of data in use at any given moment is can be as large as the number of active *neurons*, a number which can be staggeringly larger. The insight here is that the nature of the massive “bandwidth” that the brain exhibits in this way is contingent on the fact that its data is stored in a highly connected network. Such bandwidth is not possible on a standard computation model because its ratio of data visited at a given cycle to data stored is necessarily much smaller. Massive bandwidth is one important, though subtle, consequence of the representational system the brain uses—are there others?

¹At the least.

Chances are good that there are other consequences, but in order to find them we will need to begin thinking about representation less as a substance and more as a process—more as a combination of data with the means to access it. This point has been made by engineers of expert systems in the past. [6] Taking it seriously seems to be more difficult than pointing it out. For one thing, it threatens to break down a convenient abstraction barrier between data and the methods that operate on that data, something that engineers are not eager to do without significant justification.

A.2 Computational Neuroscience

While artificial intelligence is a field derived from computer science and draws inspiration from neuroscience, computational neuroscience is the opposite. People in this field seek to understand the informational capabilities of biological neurons and neural networks. To accomplish this, they study output from real neurons using electrophysiology. They build mathematical and computational models of neurons and neural networks, generally taking inspiration from systems mapped out in the brains of animals.

Computational neuroscience is useful to artificial intelligence as a toolkit of components and design principles that can be used to build intelligent systems. This is true for several reasons. CN generates theories about brain systems that are on a scale smaller than those systems that AI theorizes about, creating an excellent opportunity for collaboration. One of the criteria for success in CN is biological plausibility and this property thus transfers over to an AI system that uses CN components. Additionally, from an engineering perspective, maintaining a criteria of biological plausibility is useful because it provides more design space constraint. Moreover, the biology seems to work rather well, thus understanding what makes a system more or less plausible is equivalent to understanding what makes the biology work the way it does.

Another important perspective CN brings to AI is its embrace of systems that use neuron-plausible representations (i.e. vectors and matrices of real numbers to

represent bundles of neurons). While such systems are likely not a “silver bullet” of artificial intelligence, the light they shed on the kinds of computations that neural systems may be capable of could serve as an important constraint on the design space of AI. An important contribution CN is already making is its focus on computation as a dynamical system. In a dynamical system, you represent computations by a set of differential equations rather than a logical program. Such systems impose restrictions on the ways information can be processed. In particular, they treat representations and the methods that operate on those representations in a much more fluid way. Representations are usually maintained as constraints between conditions, rather than static objects, which introduces concerns such as stability and results in the creation of “fuzzier” systems. These issues make some kinds of computations more difficult than on a standard computation model. However, they make other kinds of computations easier. Those algorithms that are convenient for a dynamical system to carry out should draw extra attention, and by so doing, can provide additional constraint into AI research.

One class of algorithms that dynamical systems are good at are those which involve the discovery of statistical regularity in large amounts of data. Feedforward systems using error-driven learning find effective ways to represent complex input-output transformations. Principal Component Analysis, an important method for discovering the most important statistical features of a stream of data, is straightforward using dynamical systems. One property of these systems that is subtle but important, is that their representations are frequently their own best description. After training, the hidden layer of a feedforward neural network, a representation of some of the regularity in the input-output relationship of the network, is frequently difficult to categorize in terms of the input or the output. While it may be frustrating from an engineering perspective to be unable to fully interpret the system in a cell-by-cell reductionist basis, some feel that this demonstrates the importance of such models. [17] This property causes us to focus on understanding these representations as part of a larger system of the constraints that caused them to come about. This shift away from representations in favor of the system that creates them may be an

important way to analyze the activities of large populations of neurons at a higher level of abstraction. If we can replace a stream of data with a few rules that made them come about, it seems like we have better tools, not worse. David Marr calls systems who are their own best description Type II theories, and while they should be treated carefully, he feels they can still make important contributions to AI. In particular, Marr believed that type II theories may be most appropriate in areas such as low-level sensory processing—one of the areas that computational neuroscience is most interested in. [13]

A.3 Systems Neuroscience

Functional Magnetic Resonance Imaging and its cadre of related brain imaging techniques have provided a new dimension of insight onto how brains function. One of the most basic findings from this research has been the observation that as you think about things, predictable parts of your brain become activated. Predictable, in this case, means that researchers have an ever increasing understanding of how stimulus relates to brain activation in specific areas of the brain. Areas have been identified as contributing to the processing of categories of stimuli, such as faces, places, and tools.

What we see in these images are diverse clusters of brain areas activated as different mental processes occur. There is a growing understanding in the neuroscience community that the best metaphor for the way these activations are organized is as a network. [9]. Thus, depending on what your brain is doing at a given moment, different networks of brain regions will be activated, exchanging information, and causing activity.

We know more about the structure of the brain and the way it processes information than we sometimes think. While our understanding is still limited, the brain sciences are rapidly creating footholds that AI could use to generate its theories. For example, the following diagram illustrates a sketch of the broad areas of the cerebral cortex and their interconnectivity. The two most useful ideas here are the notion of

the cortex as having a hierarchical organization, and the notion of there being separate hierarchies for sensory input and motor output that interact at arbitrary levels of the hierarchy.

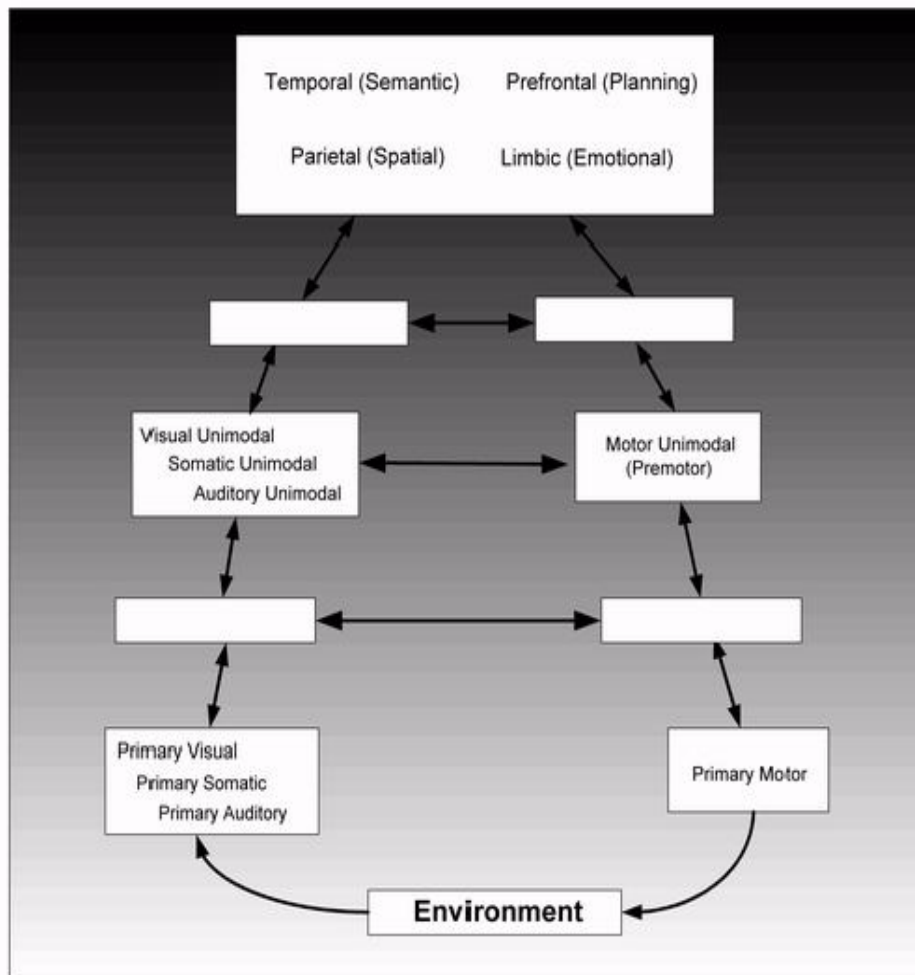


Figure A-1: The Perception-Action cycle. (from [9], modified without permission)

This hierarchical organization is not absolute. The brain has elements of heterarchy as well, allowing cross-cutting connections to be formed between layers. However seeing that an architecture of the cortex is beginning to emerge, it makes a lot of sense to allow this information to shape AI theories.

With these points in mind, it may make sense to think of a symbol as a “network of activation” that operates on top of this architecture. Such networks could be formed between or within any of the regions in figure A-1 and could have a large

variety in their content and complexity. A single network could correspond to large assemblies of neurons and the connections between them. Much of the inspiration for such networks as representational units is present in neuroscience today. [9]

A.4 Machine Learning

The field of Machine Learning, a member of the artificial intelligence family, concerns itself with two kinds of learning, supervised and unsupervised. Supervised learning is the study of systems that will categorize patterns that are presented to it in the presence of an error signal. One generally thinks about such systems as a student in the presence of a teacher who is responsible for showing it patterns and telling it what their names are. Such systems are trained on a subset of all the patterns it will ever see, and are considered useful if they can correctly classify novel patterns in a way that is consistent with its training set.

Unsupervised learning, on the other hand, is the study of systems that categorize patterns presented to it in the absence of an error signal. Such systems can be thought about as stumbling about in the dark, bumping into data, and trying to group the related data together. Such systems label data by the groups of data they can form, rather than by a teacher's instruction.

The field of data mining is a business application of unsupervised learning ideas. Its goal is to try and find correlations between large amounts of data that were previously obscure. For example, a data mining system might be set on a customer database for a large corporation, and might reveal a correlation between purchasing habits and season.

From a symbol grounding perspective, unsupervised learning is appealing. While supervised learning imposes a system of meaning by specifying an input-output relationship, unsupervised learning allows a system of meaning to be discovered based on regularities inherent in the data. Unsupervised learning is possible in the absence of extra knowledge about the data, whereas supervised learning is not. Thus if our goal is to enable a system to be able to gather information from scratch, it seems un-

likely that supervised learning alone will provide us a stable foundation to build on. Each supervised learning system will require another supervised learning system to establish the desired input-output relationship for the previous one, and the systems will chain in such a way forever.

Both supervised and unsupervised learning are thought to be present in some form in the brain, and connectionist models of the cortex have been created that incorporate both. [16]

Bibliography

- [1] P.E. Agre and D. Chapman. Pengi: An implementation of a theory of activity. In Morgan Kaufmann, editor, *Proc. Sixth National Conference American Association for Artificial Intelligence*, All ACM Conferences, pages 268–272. American Association for Artificial Intelligence, 1987.
- [2] Joachim De Beule, Joris Van Looveren, and Willem Zuidema. Grounding formal syntax in an almost real world. AI-Memo 02-03, Vrije Universiteit Brussel, Artificial Intelligence Laboratory, 2002.
- [3] Mark H. Bickhard and Loren Terveen, editors. *Foundational Issues in Artificial Intelligence and Cognitive Science*. Number 109 in Advances In Psychology. North-Holland, 1995.
- [4] B.J Copeland. What is artificial intelligence? On the WWW at http://www.alanturing.net/turing_archive/pages/Reference%20Articles/what_is_AI/What%20is%20AI06.html, May 2000.
- [5] Silvia Coradeschi. Anchoring symbolic object descriptions to sensor data. Linkoping University Electronic Press, 1999. Problem Statement.
- [6] Randall Davis, Howard Shrobe, and Peter Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993.
- [7] M. Dittenbach, D. Merkl, and A. Rauber. The Growing Hierarchical Self-Organizing Map. In S. Amari, C. L. Giles, M. Gori, and V. Puri, editors, *Proc*

of the *International Joint Conference on Neural Networks (IJCNN 2000)*, volume VI, pages 15 – 19, Como, Italy, July 24. – 27. 2000. IEEE Computer Society.

- [8] Gary L. Drescher. *Made-up Minds*. MIT Press, Cambridge, Massachusetts, 1991.
- [9] Joaquin M. Fuster. *Cortex and Mind: Unifying Cognition*. Oxford University Press, 2003.
- [10] Stephen Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.
- [11] Teuvo Kohonen. *Self-Organizing Maps*. Number 30 in Springer Series in Information Science. Springer, third edition, 2001.
- [12] Teuvo Kohonen and Riitta Hari. Where the abstract feature maps of the brain might come from. *Trends in Neurosciences*, 22(3):135–139, March 1999.
- [13] David Marr. Artificial intelligence – a personal view. AI Memo 355, MIT AI Lab, mar 1976.
- [14] Alex Martin and Linda L. Chao. Semantic memory and the brain: Structure and processes. *Current Opinion in Neurobiology*, 11(2):194–201, April 2001.
- [15] Marvin Minsky. *The Society of Mind*. Simon and Schuster, New York, 1988.
- [16] Randall C. O’Reilly and Yuko Munakata. *Computational Explorations in Cognitive Neuroscience*. MIT Press, Cambridge, Massachusetts, 2000.
- [17] David A. Robinson. Implications of neural networks for how we think about brain function. *Behav. Brain. Sci.*, 15:644–55, 1992.
- [18] H. Charles Romesburg. *Cluster Analysis for Researchers*. Lifetime Learning Publications, Belmont, California, 1984.
- [19] Deb Roy and Alex Pentland. Learning words from sights and sounds: A computational model. *Cognitive Science*, 26(1):113–146, 2002.

- [20] Terry Winograd. Procedures as a representation for data in a computer program for understanding natural language. MIT AI Technical Report 235, MIT, Feb 1971.
- [21] Patrick. H. Winston. *Artificial Intelligence*. Addison-Wesley, Reading, Massachusetts, third edition, 1993.