Applying Category Theory to Improve the Performance of a Neural Architecture

Michael J. Healy, Richard D. Olinger, Robert J. Young, Shawn E. Taylor, Thomas P. Caudell, and Kurt W. Larson

April 13, 2009

NOTICE: this is the author’s version of a work that was accepted for publication in Neurocomputing. Changes resulting from the publishing process, such as editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. The article is now in press.

Abstract

A recently-developed mathematical semantic theory explains the relationship between knowledge and its representation in connectionist systems. The semantic theory is based upon category theory, the mathematical theory of structure. A product of its explanatory capability is a set of principles to guide the design of future neural architectures and enhancements to existing designs. We claim that this mathematical semantic approach to network design is an effective basis for advancing the state of the art. We offer two experiments to support this claim. One of these involves multispectral imaging using data from a satellite camera.

Keywords

Category theory, mathematical semantics, ART 1, stack intervals, multi-spectral imaging.

1 Introduction

This paper describes an experimental test of a novel theory of knowledge representation in neural networks. Many previous studies [1, 2, 3, 4, 7, 12, 15, 13, 21] have addressed the knowledge representation capability of neural networks, but the theory behind the present study is based upon a form of mathematics with a unique explanatory capability for knowledge representation. Understanding neural computation as the acquisition, representation and use of knowledge clarifies its semantics by declaring symbolically what the computations represent. How are objects and events represented in a neural network’s connection

---

*This work was supported in part by Sandia National Laboratories, Albuquerque, New Mexico, under contract no. 238984. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy’s National Nuclear Security Administration under Contract DE-AC04-94AL85000.

†M. Healy, R. Olinger, S. Taylor, and T. Caudell are with the Department of Electrical and Computer Engineering and R. Young is with the Department of Computer Science, University of New Mexico, Albuquerque, New Mexico 87131 (e-mail: mjhealy@ece.unm.edu; rolinger@ece.unm.edu; ryoung@cs.unm.edu; shawnt@unm.edu; tpc@ece.unm.edu).

‡K. Larson is with Sandia National Laboratory, Albuquerque, New Mexico (e-mail: kwlarso@sandia.gov).
weights? How do we understand connectionist learning, generalization, and specialization in terms of data and prior knowledge? A systematic means of addressing these questions can serve as a fundamental and comprehensive base for analysis and design provided that the understanding is accompanied by mathematical rigor. But this begs the question of whether there is a form of mathematics that can serve as a vehicle for addressing questions about neural network semantics. We introduce a form of mathematics that serves as such a vehicle along with an experiment testing this claim.

The semantic theory for neural networks discussed here is based upon category theory, a form of mathematics formerly thought to be purely abstract. The theory brings mathematical rigor to the analysis of what, where and how knowledge is acquired, organized, stored, and used in connectionist systems. As will be shown, knowledge can be organized into a parallel, distributed structure of interrelated packets, or modules, that describe properties, relationships, and behaviors of different kinds or systems of items, events, and processes. Connectionist systems, like the interrelated knowledge modules, have a parallel, distributed structure. This similarity can be exploited to gain an understanding of their semantics. This can be accomplished by examining the distributed structure of knowledge, and then studying the possible ways in which the knowledge structure can be represented in connectionist structures. Category theory has a unique advantage relative to other forms of knowledge representation because its very purpose is the analysis of distributed structures and the representation of one structure within another via structure-preserving mappings.

The structure of knowledge has two main characteristics: It is a distributed system, with separate modules of knowledge describing systems of quantities; and it is hierarchical, with modules arranged from the abstract to the specific. Consider knowledge about two systems of physical quantities: electrical circuits and chemical compounds. The different entities—circuits and circuit components in one case, elements and compounds and conditions governing their reactions and equilibria in the other—interact in many different ways. Chemical bonds are electrical in nature, and electrolysis has long been used to effect chemical reactions; electrical circuits are built upon the properties of chemical substances, for example, in the form of batteries and solid-state components and conductors. The common meeting-ground of electricity and chemistry is in the abstractions of physics, the quantum states and large-scale static and dynamic properties of electrons. Knowledge about electrical circuits and chemical compounds specialize electron physics, adding details such as how electricity is made to flow in specific circuits and devices, and how chemical substances react under different conditions to form other substances. It is the abstractions that determine the interaction of electricity and chemistry. The view of knowledge that emerges is that of many, interconnected parcels, or theories. The more complex, specialized theories inherit basic knowledge from the more abstract theories. A theory can inherit from more than one more abstract theory, and it can in turn be inherited by more than one more specialized theory.

The semantic theory builds on these and other ideas to explain knowledge acquisition and representation in a neural network as an incremental re-use of an existing partial representation of the knowledge hierarchy to form a more complete representation. This process begins with concepts at or near the sensor level that express basic knowledge about the inputs, and re-uses concept representations progressively to derive new representations of more abstract and also more specialized concepts based upon the current stimuli.

We present two experiments showing that these ideas can be harnessed to advance the state of the art in neural network design. In both experiments, we compare the performance
Applying Category Theory ... Healy, Olinger, Young, Taylor, Caudell, and Larson

of the generic ART 1 network [5] with a new network, called LimitsART 1, derived from ART 1 by applying neural design principles from the categorical semantic theory. These networks separate binary input patterns into clusters representing classes of items in an unsupervised fashion. Because the data for both experiments consists of patterns of continuous values, we equip both ART 1 and LimitsART 1 with stack interval networks to convert continuous values into binary patterns having the form of a “thermometer code” along with the complementary bit pattern, a form of encoding which we refer to as “complement code” ([14, 15]). The first experiment is a test to compare the templates formed by both networks on two-dimensional patterns with points generated from a random distribution. The second experiment involves data from a satellite camera which is used by both networks to generate multispectral images. Although the data originate from an actual test of the camera, it is important to note that experimentation, and not multispectral imaging per se, is the focus of this paper. The objective is simply to determine whether category theory can be used to improve the performance of a neural architecture. Nevertheless, we hope that this is of interest to many in electrical engineering and other fields involving this kind of data. Both experimental results suggest that the modification derived from category theory improves the ART 1 network’s performance on the data.

In reference to the experiments, a quick note about terminology is in order. “Continuous”, “graded” and “grey-scale” are terms often used in certain contexts for computer representations of real-number values. We shall hereafter use the term “real values”, except that “graded” and “grey-scale” will be used where this seems to convey information about how the numerical values are being used.

Because the semantic theory is mathematically based in category theory, which is as yet unfamiliar to many (once regarded as the ultimate in pure mathematics), we begin with a brief overview of the topics necessary for understanding the work described here. Our semantic theory in its current state of development is described in full in [17]. Our previous work in applying the semantic theory to neural network analysis and design is described in, for example, [16]. The multispectral experiment described here was first presented in preliminary form in [19], but the description of the neural architecture and its design rationale was incomplete because it had to be brief. The present paper covers these topics in full. Other applications of category theory exist in physical and computational theory [9, 11, 12, 24, 26, 27, 28] and practice [20, 29].

The paper is organized as follows. Section 2 provides a very brief grounding in the category theory used. In Sections 3 and 4 we discuss knowledge, neural structure, and the mapping of the one into the other. Section 5 explores stack interval networks and real-valued data in the context of knowledge representation, and Section 6 shows how these ideas are applied to LimitsART 1. Section 7 describes the first experiment, Section 8 describes the second, and Section 9 is the Conclusion.

2 Category Theory: A Brief Introduction

Introductions to category theory at varying levels of detail are available in [8, 22, 23, 25] and [17]. It is a theory of mathematical structure based upon the notion of an arrow, or morphism—a relationship between two objects in a category. A morphism \( f : a \rightarrow b \) has a domain object \( a \) and a codomain object \( b \), and serves as a sort of directed relationship between \( a \) and \( b \). In a category \( C \), each pair of arrows \( f : a \rightarrow b \) and \( g : b \rightarrow c \) (where the codomain \( b \) of \( f \) is also the domain of \( g \) as indicated) has a composition arrow
$g \circ f : a \to c$ whose domain $a$ is the domain of $f$ and whose codomain $c$ is the codomain of $g$. Composition is associative, that is, for three arrows of the form $f : a \to b$, $g : b \to c$ and $h : c \to d$, the result of composing them is order-independent, with $h \circ (g \circ f) = (h \circ g) \circ f$. For each object $a$, there is an identity morphism $\text{id}_a : a \to a$ such that for any arrows $f : a \to b$ and $g : b \to a$, $\text{id}_a \circ g = g$ and $f \circ \text{id}_a = f$. A familiar example of a category is one called $\text{Set}$, which has sets as its objects, functions as its morphisms, and whose composition is just the composition of functions, $(g \circ f)(x) = g(f(x))$.

Key notions for the theoretical background of this paper are commutative diagrams and initial and terminal objects. A diagram is a collection of objects and morphisms of $C$. In a commutative diagram, any two morphisms with the same domain and codomain, where at least one of the morphisms is the composition of two or more diagram morphisms, are equal. An initial object, where one exists in $C$, is an object $i$ for which every object $a$ of $C$ is the codomain of a unique morphism $f : i \to a$. A terminal object $t$ has every object $a$ of $C$ as the domain of a unique morphism $f : a \to t$.

An important use of these key notions is in the definition of limits and colimits. In [16] and [17] we have shown how colimits model the learning of more complex concepts through re-use of simpler concepts already represented in the connection-weight memory of a neural network. In [17] we show how limits model the learning of simpler, more abstract concepts through re-use of existing representations. Reversing the arrows and substituting “initial” for “terminal” and “cocone” for “cone” in the following description of limits provides an overview of the dual notion, colimits. Let $\Delta$ be a diagram in a category $C$ as shown in Fig. 1, with objects $a_1, a_2, a_3, a_4$ and morphisms $f_1 : a_1 \to a_3$, $f_2 : a_2 \to a_3$, $f_3 : a_1 \to a_4$, $f_4 : a_2 \to a_4$. The diagram $\Delta$ extends $\Delta$ to a commutative diagram by adding a cone $K$, consisting of an apical object $b$ and morphisms $g_i : b \to a_i$ ($i = 1, \ldots, 4$) (called colimit leg morphisms) such that $f_1 \circ g_1 = g_3 = f_2 \circ g_2$ and $f_3 \circ g_1 = g_4 = f_4 \circ g_2$ (provided...
additional objects and morphisms with the requisite properties exist in \( C \). Cones for \( \Delta \) are the objects of a category \( \text{cone}_{\Delta} \) (whose morphisms are described in [17]). A limit for the diagram \( \Delta \) is a terminal object \( K \) in \( \text{cone}_{\Delta} \), in which case \( \Delta \) is called the defining diagram for the limit and \( \Delta \) is called its base diagram.

The importance of category theory lies in its ability to formalize the notion that things that differ in substance can have an underlying similarity of “structural” form. A mapping between categories that preserves compositional structure, called a functor, formalizes this notion. A functor \( F : C \rightarrow D \) associates to each object \( a \) of \( C \) a unique image object \( F(a) \) of \( D \) and to each morphism \( f : a \rightarrow b \) of \( C \) a unique morphism \( F(f) : F(a) \rightarrow F(b) \) of \( D \), and is such that (1) for each composition \( g \circ f \) in \( C \), \( F(g \circ f) = F(g) \circ F(f) \); (2) for each object \( a \) of \( C \), \( F(\text{id}_a) = \text{id}_{F(a)} \). It follows that \( F \) maps commutative diagrams of \( C \) to commutative diagrams in \( D \). This means that any structural constraints expressed in \( C \) are translated into \( D \). Natural transformations unify the images of morphisms under different functors. They will not be discussed here, but it is important to mention them because they fill important roles in the semantic theory and can be used in multiregion network design[17].

3 A Category for Neural Network Semantics

The semantics of a distributed system can be expressed as a system of knowledge modules describing the functions performed by each part of the system. The knowledge modules are symbolic concepts - descriptions of possible worlds, systems, and situations, real or imagined, abstract or specific. In this context, connectionist learning is the incremental representation of knowledge. The knowledge representation capability of a neural network lies in its connection structure and operational rules for stimulus response and weight adaptation, or learning. Knowledge is acquired as the network adapts. This process begins with a predetermined system of concepts at and near the sensor/actuator level that describe basic properties of inputs, outputs and internally-generated stimuli. The network learns by re-using the sensor-represented concepts in many ways in combination with the input data to “discover” concepts not yet represented by the connection-weight array of the network. In the semantic theory, category theory provides a mathematical model of this process of knowledge representation within a neural network.

We express knowledge mathematically as a category \( \text{Concept} \), whose objects are concepts and whose morphisms are similar to “sub-concept” relationships. This category is familiar to categorical logicians as a category of formal logic theories ([8, 11, 24], and see [17]). A theory morphism \( s: T \rightarrow T' \) (if one exists with \( T \) as domain and \( T' \) as codomain) is a replacement of the symbols of \( T \) by those of \( T' \) that transforms the axioms of \( T \) into either axioms or theorems of \( T' \). The composition of morphisms by composing symbol substitutions is straightforward. This provides a mathematical expression of the compositional, hierarchical (and parallel, distributed) structure of knowledge. For example, any number system, whether the integers, reals, complex, or another system, depends upon concepts that are more abstract, representing “first principles”. A concept that applies to the integers and the reals is that of a total order relation \( \leq \), expressed in a form of formal logic called first-order predicate calculus as follows:

Theory Totord
sort E
op le: E*E -> Boolean
Axiom Reflexive is
  forall (x: E) (le (x, x))
Axiom Transitive is
  forall (x, y, z: E)
    (le (x, y) and le (y, z) implies le (x, z))
Axiom PartialOrder is
  forall (x, y: E)
    (le (x, y) and le (y, x) implies x = y)
Axiom Totalness is
  forall (x, y: E)
    (le (x, y) or le (y, x))
end

The theory Totord has a specified sort, E, and also unspecified sorts that it inherits as an object of the category Concept. A sort is a formal “container” for one type of item described in a theory. The sort Boolean and its accompanying operations and:
Boolean*Boolean -> Boolean, or: Boolean*Boolean -> Boolean, and so forth (the Boolean AND, OR, etc.) constitute the theory of Booleans, which is inherited by every theory. All the other sorts of Totord, such as E*E, are derived from E and Boolean. The relation ≤, where x ≤ y is either true or false for a given pair (x, y), is expressed as an operation symbol le: E*E -> Boolean. In a model of the theory Totord in the category Set, the sort E is interpreted as a set E with an accompanying total order structure, Boolean is interpreted as the set B of truth values \{T, F\}, and the operation symbol le expressing the relation ≤ is interpreted as a function (a Set morphism) ≤: E x E -> B from the Cartesian product E x E to B assigning a truth value to each ordered pair (r, r′) (T if r ≤ r′, F otherwise), where (r, r′) ∈ E x E.

More specific than Totord are theories that incorporate it, such as those for the different theories about numbers. A part of an arbitrary number theory appears as follows:

Theory Arbnum
  sorts Num (Note: sort Boolean is understood.)
op +: Num*Num -> Num
op times: Num*Num -> Num
op <=: Num*Num -> Boolean (the ≤ relation)
const 0: Num
(Other operations and constants ...)
(Next are several axioms giving +, -, times and 0 and 1
and the other constants their usual properties - e.g.):
Axiom Lesseqdef is
  forall (x, y: Num) ((x <= y) iff
    (exists (z: Num) ((y = x + z)
      and pos (z))))
(and so forth ...)
Applying Category Theory ... Healy, Olinger, Young, Taylor, Caudell, and Larson

Theory morphisms (our concept morphisms) relate theories to other theories that inherit their properties. For example, Arbnum inherits the order relation whose properties are expressed in Totord. The morphism expressing this inheritance is given by the following symbol mapping together with the requirement that the axioms of Totord are mapped to axioms or theorems of Arbnum by symbol substitution:

Morphism Order-to-Num

\[
\begin{align*}
E & \mapsto \text{Num} \\
\leq & \mapsto \leq
\end{align*}
\]

In Section 5, we discuss the theoretical background as applied to the experiments described in this paper. This requires concepts and morphisms that express the representation of real values by binary input patterns. Binary input patterns are required as input for an ART 1 network, and this combination of architecture and input values (that is, binary) provides a convenient laboratory case for the experiments. To express the real-to-binary conversion, we devise concepts that describe quanta, or intervals, of numerical values. The most basic concept will be given the generic name StimVal, for “stimulus value”; its use will be described in Section 5:

Theory StimVal

\[
\begin{align*}
\text{import } & \text{Arbnum} \\
\text{sort } & \text{Stim} \\
\text{op } & \text{value: } \text{Stim} \rightarrow \text{Num} \\
\text{const } & \text{lev0: } \text{Num} \\
\text{const } & \text{delta-lev: } \text{Num} \\
\text{const } & \text{st: } \text{Stim}
\end{align*}
\]

The \text{import} statement is a syntactic device declaring that the theory Arbnum is a part (a sub-theory) of StimVal and there exists an import morphism which simply maps its symbols to themselves in StimVal. Thus, StimVal uses the axioms of Arbnum directly as axioms of its own. Import morphisms expressed in this way greatly simplify the writing of theories. The operation symbol value in StimVal represents the assignment of numerical values, or magnitudes, to the stimuli represented by the sort Stim. The constant st of sort Stim represents a specific but otherwise arbitrary stimulus.

4 Neural Categories and Model Categories

Each neural network architecture \(A\) and connection weight array \(w\) has an associated category \(N_{A,w}\). This category is a mathematical representation of the structure comprising the network connections, the weights in \(w\), and the potential activation states initiated by inputs given \(w\). The first design principle derived from the semantic theory is that a viable architecture and weight combination is one for which there is a functor \(M:\text{Concept} \rightarrow N_{A,w}\) mapping concepts and their morphisms to the objects and morphisms of \(N_{A,w}\). This serves as a mathematical vehicle for studying the association of knowledge with \(A\) in the state of learning associated with \(w\).
4.1 Neural Objects and Morphisms

An object of $\mathbf{N}_{A,w}$ is defined by a pair $(p_i, \eta)$, where $p_i$ is a node of $A$ and $\eta$ is a set of output values for $p_i$. Each output value in the set $\eta$ is associated with one or more input patterns for $A$ that cause $p_i$ to generate that output value given the weights in $w$; each such input pattern is thusly associated with the pair $(p_i, \eta)$. The set $\eta$ is often modeled as an interval of real values where $p_i$ has a real-valued signal function. However, this is not a requirement of the semantic theory; if complex numbers are used, for example, $\eta$ is a region in the complex plane. A morphism $m: (p_i, \eta) \rightarrow (p_j, \eta')$ of $\mathbf{N}_{A,w}$ is defined by a set $\Gamma$ of connection paths which share common source and target nodes $p_i$ and $p_j$, together with the specified output sets $\eta$ and $\eta'$ and also including a specified output set for each intermediate node in the paths of $\Gamma$. Fig. 2 shows a path set containing two paths with source $p_1$, target $p_4$, and two connections per path through intermediate nodes $p_2$ and $p_3$. A combination of path set $\Gamma$, output sets $\eta, \eta', \ldots$, and weight array $w$ has an associated set of inputs for which the nodes of $\Gamma$ produce outputs varying within $\eta, \eta', \ldots$, regardless of the fact that the activity may cause some weights to change. If $A$ is properly

Figure 2: The shared instances of the two signal paths from $(p_1, \eta)$ to $(p_4, \eta^{(3)})$ along $c_1$, $c_3$ and $c_2$, $c_4$ through the intermediate objects $(p_2, \eta')$ and $(p_3, \eta'')$ define a morphism $m: (p_1, \eta) \rightarrow (p_4, \eta^{(3)})$. There are also four morphisms $m_1$, $m_2$, $m_3$, $m_4$ given by the single-connection paths through $c_1$, $c_2$, $c_3$, $c_4$, respectively, along with the objects $(p_1, \eta), (p_2, \eta'), (p_3, \eta''), (p_4, \eta^{(3)})$. These form a diagram. If the two paths share all their instances, the diagram commutes, $m_3 \circ m_1 = m_4 \circ m_2$. 
designed and \( w \) is an array of weight values acquired at some stage of learning, it will be possible to define a functor \( M: \text{Concept} \rightarrow \mathbb{N}_{A,w} \). This is a mathematical description of the representation of concepts and their morphisms in \( A \) given \( w \). Notice that an input associated with a morphism causes a transition from \( \mathbb{N}_{A,w} \) to a different category \( \mathbb{N}_{A,v} \), representing a different stage of learning, as the weight array transitions from \( w \) to \( v \). This calls for a new functor which differs from \( M \) as determined by the weight changes.

### 4.2 Model-Space Morphisms

Each concept morphism \( s:T \rightarrow T' \) has an associated model-space morphism, a functor \( \text{Mod}(s): \text{Mod}(T') \rightarrow \text{Mod}(T) \). Here, \( \text{Mod}(T) \) and \( \text{Mod}(T') \) are categories of models, possible worlds or instances within which \( T \) and \( T' \) hold, respectively. Since \( \text{Mod}(s) \) reverses the direction of \( s \), each instance of \( T' \) has a corresponding instance of \( T \). This fact has great significance for neural networks. To see this, suppose that \( (p_i, \eta) \) and \( (p_j, \eta') \) are the images of objects \( T \) and \( T' \) under the functor \( M \), \( (p_i, \eta) = M(T) \) and \( (p_j, \eta') = M(T') \), and that \( m:(p_i, \eta) \rightarrow (p_j, \eta') \) is the image of \( s:T \rightarrow T' \), \( m = M(s) \). We associate the activating inputs for the objects \( (p_i, \eta) \) and \( (p_j, \eta') \) with objects in the model categories \( \text{Mod}(T) \) and \( \text{Mod}(T') \), respectively. Given this association, every input that activates \( (p_j, \eta') \) must also activate \( (p_i, \eta) \), a consequence of the existence of the model-space morphism \( \text{Mod}(s): \text{Mod}(T') \rightarrow \text{Mod}(T) \). Let \( T \) be the apical concept of a limit cone for a diagram \( \Delta \) in \( \text{Concept} \) and let \( s:T \rightarrow T' \) be one of the leg morphisms for the limit cone. Then, \( (p_i, \eta) \) must be activated whenever \( (p_j, \eta') \) is, where \( (p_j, \eta') \) can be any object in the diagram image \( M(\Delta) \). The reverse is true for the apical object of a colimit: Every one of its instances must be an instance of its diagram.

### 5 Experimenting with the Theory: Knowledge Representation

We conducted two experiments to answer the following question: Can we apply the theoretical background presented here to improve the performance of an existing neural network architecture? Our theory is a mathematical statement about knowledge representation in neural networks. Through category theory, concept abstraction and concept specialization are expressed via limits and colimits, where concepts are the objects in a category. A given neural architecture also yields a category, and we judge the ability of the architecture to acquire knowledge by its ability to express limits and colimits via an appropriate functor. Can we apply these ideas to provide these structures in some part of a network given that they are not already present, and thereby improve performance? A positive answer, although not definitive, would serve as an indication that the theory can be applied more generally in neural network design and thereby advance the state of the art. At the very least, the results would provide feedback to the theoretical effort.

The two experimental examples illustrate the result of applying a hypothetical functor \( M: \text{Concept} \rightarrow \mathbb{N}_{A,w} \) in the re-design of a neural network architecture \( A \), where \( w \) represents an arbitrary stage of learning. The functor is only partially implemented, yet it helps determine a modification to the architecture (LimitsART 1) that apparently improves its performance. The functor will be implicit in our discussion of commutative diagrams of concepts and morphisms mapped into \( \mathbb{N}_{A,w} \), where \( A \) is either of two architectures in each experiment: ART 1 or LimitsART 1, the same ART 1 network with the addition of
category-theoretic limits and colimits. As mentioned in the Introduction, an ART 1 network performs unsupervised clustering of binary input patterns. Each cluster is associated with a class of potential inputs which are in turn related via a similarity criterion to a binary template pattern. The template is derived by the network from those of its actual inputs which it has assigned to the cluster. The template for each similarity class is associated with an $F_2$ classifier node.

The input data for the networks in each experiment consists of patterns of $n$ real values. Because of this, both ART 1 and LimitsART 1 networks are “front-ended” with an array of $n$ neural networks called stack filter networks. Each stack filter network converts a real numerical value lying between fixed lower and upper bounds into a binary pattern which we call a stack interval. The resulting composite binary pattern, consisting of the $n$ stack intervals, is then input to ART 1 or LimitsART 1. The significance of the stack interval code in comparison with other binary codes is that it maintains the relative ordering of numerical magnitudes. This ordering is expressed in the theory $\text{Totord}$, which is mapped by a morphism into the theory $\text{Arbnum}$, which in turn has been imported into the theory $\text{StimVal}$.

The binary stack interval code for each real value $v_d$ ($d \in \{1, 2, \ldots, n\}$) contains binary 1s for those stack nodes representing pre-chosen, successively larger real values (“graded” values) exceeded by $v_d$, proceeding from left to right in figs. 3 and 4. These values are assigned to the positive stack nodes, shown at the left in the two figures; positive stack nodes representing values not exceeded by $v_d$ produce a binary 0 as output. The complement nodes, at the right, encode the complementary pattern of 0s and 1s, with the 1s representing successively larger level values that exceed $v_d$. The $n$ stack interval networks for converting the $n$ real values are concatenated end-to-end so that their binary outputs form a single composite pattern for input to an ART 1 network. Each ART 1 input node is activated by one stack node for one real value.

In the first experiment (Section 7), $n = 2$ and the data pattern components are artificially generated using a pseudorandom number generator. In the second experiment (Section 8), $n = 10$ and the pattern components are brightness values for $n$ spectral bands derived from the satellite camera data. The neural network clusters the spectral band patterns to construct grey-scale images (in lieu of artificially colored images) representing multispectral images. The images are constructed separately by two stack interval/ART 1 neural networks and compared with a panchromatic camera image.

Since the stack interval and ART 1 nodes yield binary outputs, most of the neural objects $(p_i, \eta)$ can be regarded simply as nodes $p_i$. This is true also for the stimulus node of the stack interval network at the bottom of Fig. 3, since its real-valued outputs are all regarded as instances of the single sort $\text{Stim}$ of the concept $\text{StimVal}$. Hence, although many objects $(p_i, \eta)$ of $N_{A,w}$ can be associated with a given node $p_i$, we discuss it as a single object.

5.1 The Stack Interval Network

For the present, let us drop the $d$ subscript and discuss a single real value $v$. This value, the output of the stimulus input node at the bottom of Fig. 3, is forwarded to the stack nodes through the feedforward connections shown. The $N_{psn}$ positive stack nodes, with indices $0, 1, \ldots, N_{psn} - 1$, together with their corresponding complement stack nodes indexed as $N_{psn}, N_{psn} + 1, \ldots, 2N_{psn} - 1$, constitute an array of length $2N_{psn}$ of real-input, binary-
output stack nodes. An increasing sequence of graded values \( \ell_0, \ell_1, \ldots, \ell_{N-1} \) is assigned to the nodes of each stack interval network, one per positive stack node. For simplicity, let us assume that the sequence values are equally-spaced \( \delta \ell \) units apart. All feedforward connections from the stimulus input node are excitatory (+ in Fig. 3) with unit weight. All stack nodes have a zero threshold so that their activity depends entirely upon the balance of excitatory (+) versus inhibitory (-) inputs, with the single exception of node 0, whose threshold is \( \ell_0 \). The feedforward inhibitory (-) connections from stack node 0 to the other positive stack nodes have weight \( \ell_0 + \delta \ell \) and the feedforward (-) connections from all other positive stack nodes to other positive stack nodes have weight \( \delta \ell \). The inhibitory (-) connection from each positive stack node to its complement has unit weight, as do the excitatory (+) connections forming a chain from complement stack node \( 2N_{psn} - 1 \) to node \( N_{psn} \). As a consequence, positive stack node \( q \ (q \in \{0, 1, \ldots, N_{psn} - 1\}) \) requires an input from the stimulus node of magnitude exceeding \( \ell_0 + q \cdot \delta \ell \) in order to become activated. Because of the unit-weight excitatory (+) connection from the stimulus node to complement node \( 2N_{psn} - 1 \), the chain of unit-weight excitatory (+) connections proceeding toward node \( N_{psn} \), and the unit-weight inhibitory (-) connection to each complement node from the corresponding positive node, each complement node registers the complementary binary value to that produced by its positive correspondent.

Figure 3: Binary complement-coded stack nodes, positive on the left and complementary on the right. The input node at the bottom relays the input stimulus magnitude as a real output, which is sent to all positive stack nodes and the “highest” complement node, \( 2N - 1 \). The positive stack nodes represent successively greater steps in stimulus magnitude, reading from left to right (0 through \( N - 1 \)). They and their complement nodes produce binary outputs.
Figure 4: Stack nodes, positive on the left and complementary on the right. Each of the top three rows illustrates the stack network response to an input stimulus with the magnitude indicated by \( v \). The bottom three rows illustrate template patterns resulting from the the usual ART 1 ANDing of binary input patterns component by component. “Width” refers to the width of the represented interval, in binary units.

From the foregoing, it is evident that each positive stack node represents a half-interval of the form \( \ell < v \), and its complement node represents the half-interval \( v \leq \ell \). This has the effect of representing a stimulus real value \( v \) as an interval approximation \( \ell < v \leq \ell' \) for some pair of adjacent values \( \ell \) and \( \ell' = \ell + \delta \ell \) (see Fig. 4). Let the constant \texttt{delta-lev: Num} from the theory \texttt{StimVal} denote the quantity \( \delta \ell \), \texttt{lev0: Num} the quantity \( \ell_0 \), and \texttt{value (st): Num} the current stimulus value \( v \); then, stack node 0 represents the concept

Theory \texttt{StimLB0}
import \texttt{StimVal}
Axiom lb0 is
\[
\texttt{lev0 < value (st)}
\]
end

The import morphism \texttt{StimVal \rightarrow StimLB0} maps via the functor to a path set \( \Gamma_0 \) whose only member is the feedforward (+) afferent to node 0 from the stimulus node. For \( q \in \{1, 2, \ldots, N_{\text{psn}} - 1\} \), stack node \( q \) represents the concept

Theory \texttt{StimLBq}
import \texttt{StimLB(q - 1)}
Axiom lbq is
\[ \text{lev0} + q \text{ times delta-lev} < \text{value (st)} \]
end

Notice that the concept \( \text{StimLB}(q - 1) \) is imported in \( \text{StimLBq} \). The import morphism corresponds to the feedforward (-) connection from node \( q - 1 \) to node \( q \). The afferent to node \( q \) from the stimulus node also plays a role, because it provides the only source of excitatory input, without which node \( q \) could not become active. It serves as a facilitating connection for the morphism carried by the path from \( q - 1 \) to \( q \) but cannot be in the path set for the morphism because its source is not \( q - 1 \), which is the source node for the path set.

By similar reasoning, there is a chain of import morphisms \( \text{StimLB0} \rightarrow \ldots \rightarrow \text{StimLB}(q - 2) \rightarrow \text{StimLB}(q - 1) \rightarrow \text{StimLBq} \rightarrow \ldots \rightarrow \text{StimLB}(N - 1) \). As a consequence, \( \text{StimLBq} \) contains the axioms of \( \text{StimLB}(q - 1) \), which in turn, by the composition of morphisms, include those of \( \text{StimLB}(q - 2) \), ..., \( \text{StimLB0} \). This chain of morphisms expresses the successively greater levels of stimulus required to activate the successive positive stack nodes. If a stimulus satisfies the axioms of \( \text{StimLBq} \), it must therefore satisfy the axioms of \( \text{StimLB}(q - 1) \), and this continues back along the chain to the beginning, \( \text{StimLB0} \). This characterizes exactly the property of the positive stack nodes: If node \( q \) is activated (that is, if \( \ell_0 + q \cdot \delta \ell < v \)), then node \( q - 1 \) is also activated (\( \ell_0 + (q - 1) \cdot \delta \ell < v \)), and so forth. The reverse is true for the complement nodes and the concepts they represent, for these contain axioms expressing the complementary inequalities \( v \leq \ell_0 + q \cdot \delta \ell \) for the complement stack node \( \bar{q} = N_{psn} + q \) to positive stack node \( q \). For example, the highest-numbered of the \( 2N \) stack nodes—node \( 2N_{psn} - 1 \), the complement node to node \( N_{psn} - 1 \)—is associated with the following concept, similar to the concept \( \text{StimLB}(N - 1) \) associated with node \( N_{psn} - 1 \) but with the key axiom changed:

Theory \( \text{StimUB}(N - 1) \)
import \( \text{StimVal} \)
Axiom ub(N - 1) is
\[ \text{value (st)} \leq \text{lev0} + (N - 1) \text{ times delta-lev} \]
end

Regarding \( \ell_0 \) and \( N_{psn} \cdot \delta \ell \) as lower and upper bounds on the range of values \( v \), purely for notational purposes we include two “virtual stack nodes” having indices \( q = -1 \) (no positive stack nodes are active) and \( \bar{q} = 2N_{psn} \) (no complementary stack nodes are active). The top and bottom rows in Fig. 4 illustrate these two cases. The concepts associated with the complement stack nodes \( 2N_{psn} - 2 \), ..., \( N_{psn} + 1 \), \( N_{psn} \) can be constructed successively, in that order, by importing the preceding concept. Thus, they are related in the opposite order to those associated with the positive stack nodes. They form a chain of import morphisms \( \text{StimUB}(N - 1) \rightarrow \ldots \rightarrow \text{StimUB0} \), where concept \( \text{StimUBq} \) in the chain, associated with node \( \bar{q} = N_{psn} + q \) (for \( q \in \{0, 1, \ldots, N_{psn} - 2\} \)) is as follows:

Theory \( \text{StimUBq} \)
import \( \text{StimUB}(q + 1) \)
Axiom ubq is
\[ \text{value (st)} \leq \text{lev0} + q \text{ times delta-lev} \]
Figure 5: A stimulus concept colimit with base diagram $\Delta$ and defining diagram $\overline{\Delta}$. The apical object $\text{StimInt}(q, q')$ describes a stimulus interval $\ell_0 + q \cdot \delta \ell < v \leq \ell_0 + q' \cdot \delta \ell$.

end

The stack nodes are the input nodes in the $F_0$ layer of an ART 1 network. Therefore, their concepts are those associated with each segment of $F_0$. Since a pattern of $n$ real values is often described as "$n$-dimensional", the terms "segment" and "dimension" will be used interchangeably for the partition of $F_0$ (hence, of $F_1$) devoted to each stack interval. The top three rows in Fig. 4 illustrate the stack interval patterns produced by real stimulus values lying within the intervals shown. The top row illustrates the pattern for an input lying below the stack interval network lower bound $\ell_0$, where the positive stack nodes yield zero output (empty circles) and the complement stack nodes yield unity as output (filled-in circles) due to the lack of an inhibitory input from their corresponding positive stack nodes.

Model-space morphisms are highly significant in understanding the design of Limit-sART 1, to be presented in the next section. There, specific reciprocals to the connection paths representing concept morphisms will be required to represent the associated model-space morphisms. By contrast, the stack interval networks do not require reciprocal connections because the model-space morphism property is enforced by the fact that no stack nodes can be activated unless the stimulus node is active.

5.2 Multiple Dimensions, ART, and the Missing Colimits

Let $r$ be an integer with $1 \leq r \leq N_{psn}$. For each pair of integers $(q, q')$ with $-1 \leq q$ and $q' = q + r$, we can form a colimit as illustrated in Fig. 5 (notice that this includes
the extreme cases \((-1, N_{psn} - 1)\) and \((0, N_{psn})\), where there is a single stack node active). The colimit apical object \(\text{StimInt}(q, q')\) is the least complex concept that combines the pair \(\text{StimLB}q\) and \(\text{StimUB}q'\) along their common sub-concept \(\text{StimVal}\). The concept \(\text{StimInt}(q, q')\) expresses the two axioms \(\ell_0 + q \cdot \delta \ell < v\) and \(v \leq \ell_0 + q' \cdot \delta \ell\), thereby representing the interval \(\ell_0 + q \cdot \delta \ell < v \leq \ell_0 + q' \cdot \delta \ell\) (except that for \(q = -1\) the lower bound is \(\ell_0 = v\)). This represents a “window” of stimulus values of width \(r \cdot \delta \ell\) lying between \(\ell_0 + \max(q, 0) \cdot \delta \ell\) and \(\ell_0 + q' \cdot \delta \ell\).

These colimits cannot be represented explicitly in an ART 1 architecture because it provides only the \(F_1\) and \(F_2\) layers and their adaptive interconnections for colimit formation [16]. According to the semantic theory, the fact that the colimits shown in Fig. 5 cannot be represented in an ART 1 network, and also that there is no provision for limits, is a significant limitation on the capability of the stack interval/ART 1 network.

Fig. 6 shows an ART 1 network whose input layer \(F_0\) consists of stack nodes (a typical stack interval network is illustrated in Fig. 3). Each of \(n\) stack interval networks converts the real value in its component of an input stimulus pattern into a discretized form using the sequence of graded values \(\ell_i\) as mentioned earlier. The discretized value is represented by “complement code”, a binary “thermometer” code together with its complementary binary pattern, forming one of \(n\) segments of \(F_0\). In the experiments to be presented, each data pattern component is sampled using the same sequence of pre-chosen graded values (using the same values is a convenience, not a necessary restriction on arrays of stack interval networks). The ART 1 network classifies the real-valued input stimulus patterns based...
upon their stack interval-encoded binary patterns. The result of presenting the combined stack interval/ART 1 network with a number of real-valued patterns is a binary template pattern representing each class (actually, representing the \( n \)-segment binary stack interval patterns associated with each class). A two-segment binary \( F_0 \) pattern is shown in Fig. 6, with positive and complement binary stack node arrays designated \( b_d \) and \( b_d' \), respectively, for real values \( v_d \).

The same binary values are copied through one-to-one efferent connections to the \( F_1 \) layer, as shown, in the usual ART 1 fashion. The \( F_1 \) layer therefore represents the \( n \) real values via the stack interval encoding, as does \( F_0 \). We designate the part of the \( F_1 \) layer consisting of \( b_d \) and \( b_d' \) together as \( F_{1,d} \), \( d \in \{1, 2, \ldots, n\} \). For simplicity, the array of nodes in all positive stacks are given the same length \( N_{psn} \), giving each \( F_{1,d} \) a total of \( 2N_{psn} \) nodes, so that the number \( N \) of \( F_1 \) (and \( F_0 \)) nodes is \( N = 2nN_{psn} \). The vigilance sub-system of the network (with vigilance node \( V \) in Fig 6) accepts the current \( F_2 \) “choice” node to represent the current input pattern \( I \) if

\[
\frac{\|I \land T^k\|}{\|I\|} \geq \rho
\]

and rejects it otherwise, where \( I \land T^k \) is the bit-wise “AND” of \( I \) and the choice connection-weight template \( T^k \), \( \|X\| \) is the number of 1s in a binary pattern \( X \), and \( \rho \) is the vigilance parameter. Because of the multiple real values represented in \( F_1 \), the patterns \( I, T^k \) and \( I \land T^k \) are composed of sub-patterns, or pattern segments, \( I_d, T^k_d \) and \( I_d \land T^k_d \), respectively for \( d \in \{1, 2, \ldots, n\} \). Because the binary input patterns are “complement-coded” by the \( n \) stack interval networks, the templates that form can be visualized as hyperboxes in \( n \)-dimensional Euclidean space ([14, 15]). The template segment corresponding to dimension \( d \) represents the \( d \)-th side of that hyperbox, based upon the stack interval encoding and the manner of “erosion” of binary 1s in an ART 1 template. Each side of the hyperbox has a length of the form \((q' - q)'\delta\ell\), representing an interval \( \ell_0 + q \cdot \delta\ell < v \leq \ell_0 + q' \cdot \delta\ell \) of real values for the corresponding stimulus inputs associated with that component of the template pattern (except that for \( q = -1 \), \( \ell_0 = v \)). This interval can be expressed as a concept which can be calculated as the apical object of a colimit of the form \( \text{StimInt}(q, q') \) in Fig. 5. Yet, there is no provision for this colimit in the ART architecture.

Since there can be no representations in ART 1 for the \( \text{StimInt}(q, q') \) colimits, a stack interval/ART 1 network has no control over the lengths \((q' - q)'\delta\ell\) of the sides of its template hyperboxes. We proved in [14] that the stack interval/ART 1 representation generates templates equivalent to Fuzzy ART templates to within the resolution of the stack network used. However, Fuzzy ART has the same shortcoming as ART 1 in representing colimits (and limits). Therefore, neither network can control the shape of its hyperboxes, only the overall hyperbox circumference (via the vigilance system). We shall show, however, that the stack interval/ART 1 network lends itself to a modification that endows it with the capability to control hyperbox shape.

6 Limits

ART 1 Implements Limits and Colimits

The fact that the stack interval/ART 1 architecture lacks the ability to exert control over the length of each side for its template hyperboxes suggests a needed modification at or near the \( F_1 \) level. Controlling the maximum size of each hyperbox side means controlling
the individual regions $T_d^k$ of each template $T^k$ to restrict the maximum difference among all possible values $r = q' - q$. The fact that an interval $\ell_0 + q \cdot \delta \ell < v \leq \ell_0 + q' \cdot \delta \ell = \ell_0 + (q + r) \cdot \delta \ell$ is expressed in a colimit apical concept $\text{StimInt}(q, q')$ suggests a strategy: In the segment of the $F_1$ layer for each stack interval network, attach the architectural structure necessary to represent the colimit apical object $\text{StimInt}(q, q')$ (shown in Fig. 5) to each pair $q, q'$ of stack nodes having the desired separation $r_{\text{max}} = q' - q$.

6.1 The New Colimit Representations

Let us fix a particular stimulus pattern component/hyperbox dimension/segment $d$ of the ART network $F_0$ and $F_1$ layers, with $1 \leq d \leq n$. The discussion in the remainder of this section will apply separately for each value of $d$; we will consider all $n$ segments together when appropriate.

The concepts $\text{StimLBq}$ and $\text{StimUBq}'$, their common sub-concept $\text{StimVal}$, and the morphisms $\text{StimVal} \rightarrow \text{StimLBq}$ and $\text{StimVal} \rightarrow \text{StimUBq}'$ in the base diagram $\Delta$ of the desired colimit (Fig. 5) are already explicitly represented by the stack interval for segment $d$ of $F_0$, and, hence, $F_1$. To represent the colimit, we need only add a node and connections representing the colimit apical object $\text{StimInt}(q, q')$ and its cocone leg morphisms, whose domains are base diagram objects (reciprocal connections are also needed, since here we need to actively enforce the model-space morphism property). The leg morphisms $\text{StimLBq} \rightarrow \text{StimInt}(q, q')$ and $\text{StimUBq}' \rightarrow \text{StimInt}(q, q')$ are represented by afferent (+) connections from the two $F_1$ nodes representing $\text{StimUBq}$ and $\text{StimUBq}'$ together with the reciprocal connections. The leg morphism $\text{StimVal} \rightarrow \text{StimInt}(q, q')$ is represented by the set containing the two paths from the $\text{StimVal}$ node, through the stack nodes representing $\text{StimUBq}$ and $\text{StimUBq}'$, and terminating at the $\text{StimInt}(q, q')$ node, along with a reciprocal connection from the $\text{StimInt}(q, q')$ node to the $\text{StimVal}$ node. Small, positive fixed weights for these connections ensure that the colimit node requires input from both nodes $q$ and $q'$ in order to become active, thereby ensuring that the paths representing the colimit defining diagram are simultaneously active if at all. This ensures the commutativity of the defining diagram for the colimit in the category $\mathbb{N}_{A,w}$, which represents the defining diagram for the $\text{StimInt}(q, q')$ colimit in the category $\text{Concept}$.

Fig. 7 shows several colimit apical object nodes and connections added to each $F_1$ segment of the ART 1 network of Fig. 6. The nodes representing the colimit apical objects $\text{StimInt}(q, q')$ for each of the $n$ stack interval networks all together form a new layer which we call $F_1^+$. We shall continue to focus upon a particular dimension $d$, and, hence, now upon the segment $d$ of $F_1^+$. Suppose that the maximum hyperbox side length to be allowed to the resolution of the stack interval representation is $r_{\text{max}} \cdot \delta \ell$, with $1 \leq r_{\text{max}} \leq N_{\text{psn}}$. This determines several positive integer pairs $(q, q')$ with $-1 \leq q \leq N_{\text{psn}} - r_{\text{max}}$ (where all quantities are as in Section 5) and $q' = \max(q, 0) + r_{\text{max}}$. These pairs represent all possible intervals having the maximum length $r_{\text{max}} \cdot \delta \ell$ for side $d$ of any template hyperbox, corresponding to an interval $\ell_0 + \max(q, 0) \cdot \delta \ell < v \leq \ell_0 + (\max(q, 0) + r_{\text{max}}) \cdot \delta \ell$. The maximum side length $r_{\text{max}} \cdot \delta \ell$ is to be applied uniformly across the $n$ dimensions. To understand why the colimits represent maximum side lengths, suppose that a hyperbox “grows” so that side $d$ has length $r \cdot \delta \ell$ and is represented by the pair $(q_1, q_2)$ with $q_2 = q_1 + r$. Suppose further that the side exceeds the maximum length, $r > r_{\text{max}}$. This means that when the corresponding template is active, and for any of the colimit $(q, q')$ pairs, either $F_1$ node $q$ is inactive (if $q_1 < q$) or node $q' = q' + N_{\text{psn}}$ is inactive (if
Figure 7: LimitsART-1 augments the $F_1$ layer with colimits and limits. The colimit apical objects correspond to nodes in the $F_1^+$ layer and the limit apical objects correspond to nodes in the $F_{-1}^+$ layer. Each limit has the colimits for a stack interval network for one segment in its base diagram.

$q' < q_2$). Then, none of the colimit nodes in that segment of $F_1^+$ can remain active since at least one of its leg morphisms must be inactive, the nodes $q, q'$ being the sources of the connections implementing them (refer to Fig. 7).

As usual, an exception must occur for the extremes $q = -1$ and $q' = N_{psn}$. For the first case, the colimit object node in $F_1^+$ has an inhibitory (-) afferent from the positive stack node $q = 0$ and an excitatory (+) afferent from the complement node representing $\text{StimUB}q'$ where $q' = r_{\text{max}}$. The second case is symmetrical with this.

6.2 The New Limit Representations

There are several $\text{StimInt}(q, q')$ concept colimit representations for each hyperbox dimension $d$, as many as there are intervals of width $r_{\text{max}} \cdot \delta \ell$ lying within the range $\ell_0$ and $\ell_0 + (N - 1) \cdot \delta \ell$. But it is irrelevant to our purpose which of these intervals contains the actual hyperbox side $d$: Only the side length $r_{\text{max}} \cdot \delta \ell$ is important, since that is what we wish to control. This control can be exerted by applying the knowledge that exceeding the side length will result in all colimit nodes becoming inactive in segment $d$ of $F_1^+$. This calls for an abstraction: remove the information specific to each particular pair $(q, q')$, leaving only $r_{\text{max}} \cdot \delta \ell$. Since each particular pair is represented by a colimit node, the abstraction can be achieved by attaching the node and connections necessary to form a limit cone for
a diagram containing all colimit apical objects represented in segment \( d \) of \( F_1^+ \), including reciprocal connections for the limit leg morphisms (see Fig. 7).

Since there are \( n \) stack interval networks, there will be \( n \) limit cones. As shown in Fig. 7, their apical object nodes form a new layer designated \( F_{-1}^+ \), which has a single node for each of the \( n \) segments of \( F_1^+ \). The added node \( L \) is in all the base diagrams; it represents a concept \( \text{AllStimInt} \) formed from \( \text{StimVal} \) and the concepts \( \text{StimInt}[q,q'] \) via concept morphisms represented by the appropriate connections. Its inclusion is necessary for the limit cones to correctly represent Concept limits. If \( r_{\max} \leq \frac{1}{2} N_{\text{psn}} - 1 \), \( \text{AllStimInt} \) will be an inconsistent concept, since it requires \( v \) to be in all intervals at once. Inconsistent theory representations cause no harm as long as they are not given a role in network operation, and the only role for node \( L \) is to represent \( \text{AllStimInt} \). The \( F_{-1}^+ \) nodes and their attendant connections correspond to limit cones in the neural category \( N_{A,w} \) which are images of the appropriate limit cones in Concept.

Using the property of a functor \( M: \text{Concept} \rightarrow N_{A,w} \) and other categorical principles, we have redesigned the ART 1 network to obtain LimitsART 1, which enforces the properties conferred by colimits and limits from the category Concept. As long as some \( F_1^+ \) colimit object node is active in each segment \( d \), its limit object node in \( F_{-1}^+ \) will be active because of the model-space morphism property designed into the connections representing the limit cone morphisms. Since it has no other inputs, on the other hand, the limit node will be inactive if all colimit nodes are inactive. Now, as previously shown, the requirement that some colimit node in segment \( d \) be active is \( r \leq r_{\max} \), which can be rewritten \( (N_{\text{psn}} - r) \geq (N_{\text{psn}} - r_{\max}) \). Hyperboxes are often shown graphically projected in a 2D plot. Along the dimension \( d \) axis, one can imagine each positive stack node and its complement superimposed, from positions \( \ell_0 \) to \( \ell_0 + (N - 1) \cdot \delta \ell \). With this understanding, the \( d \)-th side of any hyperbox corresponds to the inactive superimposed \( F_1 \) nodes (representing template 0-bits) when the hyperbox template is active. The total number of superimposed nodes (reduced template nodes) in segment \( d \) is \( N_{\text{psn}} \). Let the quantity \( t \) be defined by \( t = (N_{\text{psn}} - r_{\max})/N_{\text{psn}} \). It represents a sort of threshold for the limit node since it is the proportion of 1-bits in segment \( d \) of the reduced template that must remain to avoid exceeding the maximum hyperbox side length. The corresponding number of 0-bits is \( 1 - t = r_{\max}/N_{\text{psn}} \). The requirement \( (N_{\text{psn}} - r) \geq (N_{\text{psn}} - r_{\max}) \) can now be rewritten in terms of the limit threshold, \( (N_{\text{psn}} - r)/N_{\text{psn}} \geq t \).

### 6.3 The Limit Representations Control Vigilance

Connections from the \( F_{-1}^+ \) nodes to the ART 1 vigilance node \( V \) through the node labelled \( S \) as shown in Fig. 7 allow each \( F_{-1}^+ \) node to supplement the vigilance node’s \( F_2 \) reset capability, as follows. First, node \( S \) is tonically active except when the entire \( F_{-1}^+ \) layer, acting cumulatively, suppresses its activation through \( n \) inhibitory (-) connections targeting \( S \), one for each \( F_{-1}^+ \) node. Therefore, if resonance between the current input and a template pattern is about to occur, but any one of the \( F_{-1}^+ \) nodes is inactive (because all of its \( F_1^+ \) correspondents are inactive), the resulting lowering of the cumulative inhibitory signal to \( S \) allows it to become active. Acting through its (+) efferent, \( S \) thereby activates \( V \), effectively vetoing the resonance via an \( F_2 \) layer reset. To avoid activating \( V \), then, each segment \( d \) must satisfy the inequality \( \| I_d \wedge T_d^k \| \geq N_{\text{psn}} - r_{\max} = t \cdot N_{\text{psn}} \).

An \( F_1 \) activity pattern \( I \wedge T^k \) is made up of the activity patterns \( I_d \wedge T_d^k \). We have assumed uniformity in representing the real values by stack interval networks: That is, all
Applying Category Theory ... Healy, Olinger, Young, Taylor, Caudell, and Larson

of the $n$ $F_1$ segments have the same number of nodes $2N_{psn}$ representing the same set of pre-chosen magnitudes, so that $N = 2nN_{psn}$ where $N$ is the number of nodes in $F_1$ (hence, also in $F_0$). The added network structure in LimitsART 1 (Fig. 7) provides for a uniform upper bound on hyperbox side length. This allows the user of the network to exercise a more specific control over template information “erosion” during re-coding than is allowed by having a vigilance parameter alone. Just as $\rho$ can be applied to control the amount of specialization versus generalization allowed in the templates, $t$ can be applied to control the specialization versus generalization allowed in each of the $n$ segments of the templates. A higher value for either parameter means smaller hyperboxes, hence, fewer input exemplars per template, and therefore greater specialization.

It is natural to ask if the parameter $\rho$ can be eliminated altogether. It can be shown that the usual test involving $\rho$ is indeed redundant if $t \geq \rho$. However, this is not the case when $t < \rho$, and therefore the parameter $\rho$ cannot be eliminated.

6.4 Two Experiments

Two experiments were designed to test whether LimitsART 1 provides an improved classification of input data when compared with that obtained with ART 1. The data domains for the two experiments were chosen to enable tests of separate hypotheses about performance improvement. In both experiments, a single LimitsART 1 simulation program was used. The simulator was programmed so that with a nonzero value of the aforementioned threshold value $t$ it simulates LimitsART 1 at that threshold, and with $t = 0$ it simulates ART 1. Finally, for full disclosure, note that Experiment 2 was performed before Experiment 1 because LimitsART 1 was developed while working with multispectral imaging data (see Section 8). The ordering of the experiments is appropriate because Experiment 1 clarifies the mechanism of the putative performance improvement, thus shedding light on the results exhibited in Experiment 2.

The stack interval/LimitsART 1 simulator was produced using our recently-developed network specification and simulation package eLoom[6]. A major advantage of working with variations on ART 1 is that, while the theory behind it is substantial and mathematically guarantees that it performs classification in a well-defined manner for items represented by binary input patterns [5], its simulation involves a very simple algorithm. LimitsART 1 simulation is equally simple. This simplicity can be attributed directly to the mathematical performance guarantees of the theories behind ART 1 (see, for example, [10]) and LimitsART 1 (see, for example, [18]). In particular, a choice of colimits, all representing intervals with the same length for all segments of $F_1$, determines the maximum allowed hyperbox side length. The choice is specified as the spacing of nodes in the colimit base diagrams, given by the threshold parameter $t$ for the activation of limit apical nodes. Thus, an abstraction derived from the semantic theory (limits built upon colimits) has given rise to a numerical parameter.

7 Experiment 1: Can Well-Separated Clusters be Well Separated?

Experiment 1 tests the hypothesis that LimitsART 1 can yield improved performance over ART 1 on the type of problem it is meant to address. Is the control of hyperbox side length a reliable way to generate hyperboxes for clusters that are naturally separated into hyperbox
regions with fixed side lengths in each dimension? To make this as simple and unequivocal a test as possible, the experiment data is separated into clusters which are circumscribed by 2-dimensional square hyperbox regions, corresponding to the use of a single threshold $t$ across all dimensions. The minimum separation between clusters is equal to the side length, and this separation occurs in both dimensions (see figures 8 and 9). Testing the hypothesis serves two purposes: First, it illustrates what in general is to be gained by controlling hyperbox size. Second, it provides a clear exercise with the minimal expected difficulty in obtaining this gain: Forming the appropriate hyperboxes for clusters as well-separated as described must be a minimum requirement for improvement.

7.1 Experimental Procedure

The neural network data were constructed to represent points in 2-dimensional Euclidean space, with a natural spacing into four clusters separated as suggested. Each point was represented by a real-valued pattern of $n = 2$ components, corresponding to two $F_0$ stack interval segments representing two hyperbox dimensions. A square rectangular region was defined, and 400 data points generated within this region as follows. First, four square rectangular subregions were defined within the larger region to serve as a base for defining four well-separated clusters. Each cluster was contained within one of the subregions. The larger region had min/max coordinates 0.0/1.0 in both dimensions. The four smaller regions were defined to have min/max coordinates at 0.2/0.4, 0.6/0.8 in the two dimensions, resulting in four square rectangular subregions, one for each cluster (see figures 8 and 9). The resulting separation distance in each dimension between the four cluster regions was 0.2 in at least one coordinate, equal to the length of each side of each subregion. Thus, the minimum spacing between clusters was as great as the width of a cluster in either of the two dimensions. The points in each of the four clusters were generated by applying a pseudorandom number generator to obtain 100 points according to a uniform distribution within the appropriate lower/upper bounds in each dimension. This resulted in a total of 400 points for the four clusters. The clustering problem for both ART 1 and LimitsART 1 was to discover through adaptation to the 2-component input patterns the network weights that provide four templates, hence, four hyperboxes, one hyperbox to bound each of the four clusters.

LimitsART 1 (where the simulator is given the threshold value $t > 0$, bringing the colimits and limits into play in the network architecture) with stack intervals restricts each side of its 2-dimensional hyperboxes to within the same allowed maximum length. Therefore, it would be expected to separate the four clusters exactly with the appropriate threshold value, $t = 0.8$, as long as the vigilance value $\rho$ does not exceed 0.8. Since $1 - t = 0.2$, this threshold value allows a hyperbox to have a half-circumference (sum of lengths of the left and lower sides) of up to 0.4 units while restricting each side to a maximum length of 0.2. Therefore, a hyperbox cannot encompass points in more than one of the four well-separated clusters, yet is allowed to encompass all the points in a cluster. Generic ART 1 ($t = 0$) with stack intervals has no such guarantee. With a vigilance value of $\rho = 0.8$, it generates hyperboxes which distribute the value 0.2 over all sides cumulatively; that is, a hyperbox might have one side at a length of 0.1 and the other at a length of 0.3, for a total half-circumference of 0.4. Thus, a hyperbox might encompass points in two of the specified clusters while covering each cluster only partially.

With lower threshold settings, $0 < t < 0.8$, LimitsART 1 would be expected to
yield partial success; this effect might be enhanced by specifying values of $\rho$ such that $t < \rho < 0.8$, allowing the ART vigilance subsystem to provide a check on overall hyperbox size. Parameter combinations such as these might well be used on arbitrary data sets, where the appropriate settings (if any) are not known and clustering with experimental parameter variation might be useful in a search for a combination yielding acceptable performance. Generic ART 1 ($t = 0$) with $0 < \rho < 0.8$ would be expected to perform less well. If either parameter $t$ or $\rho$ exceeds 0.8, the points in each of the four clusters would be divided among several hyperboxes. This would obviate the identification of the correct clusters with single hyperboxes. Generic ART 1 might still generate hyperboxes encompassing points in more than one cluster unless $\rho \geq 0.9$. In summary, it was expected that in Experiment 1, with its known, well-separated clusters, the generic architecture could (and often would) produce non-square hyperboxes that do not separate the four clusters appropriately ($\rho < 0.9, t = 0$) or else produce small hyperboxes that “fracture” the four clusters into smaller clusters ($\rho \geq 0.9, t = 0$). Since the sole objective was to test the LimitsART 1 architectural modification, other clustering methods were not considered.

Several combinations of parameter values $\rho, t$ were tried, with 20 runs made for each combination. The following procedure was used to obtain the input data for each combination over the 20 runs. First, the aforementioned procedure was used to obtain a new data set having four clusters each with 100 randomly-located points lying within a hyperbox boundary as previously described. This data set was used for all 20 runs for that parameter combination. For each run, all 400 points from the data set were input to the stack interval/ART 1 network simulator for clustering. The order in which the points were input was randomized anew for each run, and the points were input in the same order for each of three passes through the data. (Only two passes are necessary to stabilize the clusters for

Figure 8: Cluster hyperboxes generated by ART 1 (where $t = 0.0$) with $\rho = 0.8$. 
Applying Category Theory ... Healy, Olinger, Young, Taylor, Caudell, and Larson

Figure 9: Cluster hyperboxes generated by LimitsART 1 with $\rho = 0.8$ and $t = 0.8$.

data set whose input patterns are all of a single size—see the seminal paper on ART 1 clustering by Heileman et al [10]. The third pass was used for an automated procedure that tallies the results.) The stack interval coding for converting the two real values in each input pattern into a binary pattern was as follows. The positive stack for each of the $n = 2$ real values consisted of $N_{psn} = 100$ binary bits (the notation used here is the same as in previous sections). With the complementary stacks included, this yielded a size of $N = n \cdot 2N_{psn} = 2 \cdot 2N_{psn} = 4N_{psn} = 400$ bits in each of the resulting input patterns for ART 1, which therefore had 400 input $F_0$ nodes and 400 $F_1$ nodes. For each combination of network parameters $\rho, t$, the data set with its 400 2-dimensional points was converted in this fashion by the stack interval networks “on the fly” to 400 ART 1 binary input patterns during each pass through the data.

The parameter value combinations used were selected to simulate experimentation with an arbitrary data set to find a combination yielding acceptable (loosely, “the best”) performance. The fact that the correct hyperboxes were known in advance for this experiment allowed us to develop a figure of merit for measuring performance with a single number. We call this number “Efficiency”, defined as follows. First, we analyzed the result of each run for each parameter combination to find the number “Correct Templates” for that run. A correct template is one whose hyperbox encompasses exactly the points in one of the four clusters; if a hyperbox either spans more than one cluster or is restricted to one cluster but omits some of its points, the template is incorrect. We also obtained the number “Total Templates” for that run, and then found “Efficiency” as the quotient Efficiency = (Correct Templates) / (Total Templates). Obviously, perfect performance is signified by an Efficiency value of 1.0. Finally, we defined the number “Abandoned Templates” to keep track of tem-
Table 1: Summary of Experiment 1 results. Note that the cases with \( t = 0.0 \) are results for ART 1. Nonzero values of \( t \) apply to LimitsART 1.

plates which, in the final pass through the data for one run, “lose” all the points which were assigned to the template during earlier passes

### 7.2 Results

Figures 8 and 9 show the results for one run at the indicated parameter combinations. In the result shown, stack interval/ART 1 \( (t = 0) \) with \( \rho = 0.8 \) (the optimal value for \( t \)) yields one abandoned template and one that spans two clusters; the two templates that exactly encompass the clusters on the left are correct. Thus, there are two correct templates and six total templates, with one abandoned. Table 1 shows the results for all parameter combinations used in this experiment, performed as if increasing values were being tried, with \( \rho \) always greater than \( t \) (otherwise, vigilance is completely overridden by the effect of the threshold value). The aggregate results for template counts and efficiencies are shown as means and standard deviations over the 20 runs for each combination.

Notice that for each value of \( \rho \), the results vary systematically in favor of increasing performance with increasing \( t \). All numbers—Total Templates, Abandoned Templates, Correct Templates, and Efficiency—show this trend. As either \( \rho \) or \( t \) approaches the value 0.8 from below, performance increases, with increasing \( t \) having the more dramatic effect for \( \rho \leq 0.75 \). For \( \rho = 0.8 \), the efficiency increases with \( t \) toward unity, reaching it only at \( t = 0.8 \), where the effect of \( \rho \) is completely overridden. At the two parameter combinations tried for \( \rho = 0.9 \), the efficiency drops to zero regardless of whether \( t > 0 \). Efficiency as we have defined it imposes a rather stringent performance measure, but the number Total Templates does suggest a sudden increase in the number of templates with a falling-off in performance for \( \rho \) values exceeding 0.8. This effect might be noticed with an arbitrary data set using some other criterion for performance, based upon the observed effects of clustering in the domain from which the data were obtained. In summary, the results show clearly the intended effect in an experiment designed for this purpose.
8 Experiment 2: Multispectral Imaging with Satellite Camera Data

The hypothesis of the second experiment is that supplementing ART 1 vigilance by controlling hyperbox side length can yield an improvement over ART 1 in a test with data from an application domain. In this case, the data are the result of post-processing from a satellite camera, used for multispectral imaging.

8.1 Experimental Procedure

Pixel data in the form of intensities for multiple spectral bands constituted the input data points for clustering; that is, “the pixels were clustered”. The underlying notion was that an ART 1-like similarity criterion applied to the pixel data would separate the pixels into clusters for use in generating a false-color image. The image would be generated by assigning each pixel a color uniquely assigned to its cluster template. There was no intent to assign colors to templates based upon analysis of the spectral band intensities themselves: Indeed, colors were assigned to the templates arbitrarily. Grey-scale values were used in lieu of colors for reporting the results in this paper.

The same 10 bands were sampled for each multispectral image pixel. The 10-component pattern of band intensity values for each pixel was encoded by 10 stack interval networks, each having $N_{psn} = 8$ bits in its “positive” stack representing the same 8 intensity magnitudes for each band. Each of the 10 binary pattern segments generated by the stack interval networks contributed $2N_{psn}$ bits, resulting in $N = n \cdot 2N_{psn} = 10 \cdot 2N_{psn} = 20N_{psn} = 160$ bits in the resulting input pattern for ART 1, which therefore had 160 input $F_0$ nodes and 160 $F_1$ nodes. As discussed in the next sub-section, the image pixels formed a 400 by 400 array, resulting in 160,000 binary input patterns, one input pattern for each multispectral image pixel. These 160-bit input patterns were used to train each of the two ART 1 networks (that is, the stack interval/ART 1 network simulator at threshold values $t = 0$ and $t > 0$ as in Experiment 1) in a manner similar to that of Experiment 1. The resulting clusters contained multispectral image pixels having spectral value distributions that were similar according to the ART 1 network’s criteria with the specified parameter settings for $\rho$ and $t$.

8.2 Data Source: Discussion

The automated classification of pixels in satellite imagery is difficult. A combination of environmental factors makes this problem so complex that there is no general solution, and a scientific verification of any proposed solution for a particular case is tedious and expensive. The satellite imagery pixel classification domain was chosen not because we believe that the neural network technique is competitive in that domain, but because it provides a heterogeneity in input that would be difficult to obtain in a data set created purely for experimental purposes. The experimental objective was to determine whether the semantic theory could be applied to re-design a neural network and thereby improve its performance.

The source of the experiment data was a test of a technique meant to reduce the download bandwidth in high-spatial-resolution, multispectral imaging with satellite-mounted cameras. Three data sets were made available to us, two of which we used for our experiment. The three sets were derived by algorithms (which were not made available to us)
Applying Category Theory ... Healy, Olinger, Young, Taylor, Caudell, and Larson

from a single image data set (also not made available to us) taken by a camera with very high spatial and spectral resolution, covering a large portion of the optical spectrum. An overview of this data set, which is called the “data cube”, will be given first to further an understanding of the data we were given, followed by an overview of the data sets derived from it, two of which we used.

The data cube is a mapping of a 3-dimensional cube to real values (representing spectral band intensity values), thereby forming a 3-dimensional surface in 4-dimensional space. The 3-cube \( x \) and \( y \) coordinate values are 2-d pixel \((x, y)\) locations for a single ground image shot with with the very high-spatial/spectral-resolution satellite camera. Each 3-cube \( z \) coordinate value is an index representing a narrow spectral band for a very high-resolution spectrum (from 1 to approximately 250). Each \((x, y, z)\) triple is mapped to a spectral intensity value \(w\), representing the intensity in band \(z\) at that \((x, y)\) location as recorded by the camera. The \((x, y)\) pairs represent a 2048 X 2048 pixel view of the ground, to be shown as we discuss the data sets derived from the data cube. The data cube as a whole therefore represents a mapping by which there is a vector of approximately 250 \(w\) values at each \((x, y)\) pair.

Three sets of image data that had been synthesized from the data cube through data compression were made available to us. First, a high-spatial-resolution, grey-scale image of 2048 X 2048 \((x, y)\) pixels was provided. This has the lowest attainable spectral resolution, resulting from the compression of all 250 \(w\) values at each \((x, y)\) into one of 250 grey-scale brightness values for a single “smeared” band. This we referred to as our panchromatic image. Second, an image with low spatial resolution was provided, with approximately 100 X 100 pixels but with the full 250-band spectral resolution, resulting from “averaging over a group of pixels” for each band. Third, a “compromise” image was provided with intermediate spatial and spectral resolution, resulting from the application of compression in \((x, y)\) (over “groups of pixels”) and also over bands \(z\). This produced a multispectral image data set having an \((x, y)\) range of 400 X 400 pixels in 10 spectral bands. It was these three synthesized image data sets that were made available to us. We used two of them as expedient for conducting a simple experiment. Performance was to be judged by the ability to construct a grey-scale image from the multispectral data that met a certain standard. We used the 2048 X 2048 pixel panchromatic image as the standard of success. That is, two images, one derived (as described presently) from the 10-band data via a standard neural network architecture and the other derived from the 10-band data via a putative improved architecture, would be compared against the panchromatic image, which would serve as a standard. We used the panchromatic image as a success standard in lieu of the full data cube because at the time we processed the data only the three synthesized data sets were available to us and the cube data would have required processing in any case. Because the data were to be used to test a performance improvement in neural network architecture design as opposed to benchmarking a method for multispectral imaging, and because the source of the data was highly reliable, we felt that the panchromatic image would be sufficient as a success standard.
8.3 Executing the Procedure

Several combinations of network parameters $\rho, t$ were tried, as in Experiment 1 but this time measuring success according to evaluation criteria relating to multispectral imaging—after all, proper hyperbox dimensions for pixel clusters were not known and well-separated clusters could not be expected in any case. Following the three passes through the data for a given parameter combination, each resulting template was assigned a grey-scale value. Each multispectral image pixel was assigned the grey-scale value of the ART 1 class template with which it was associated. These grey-scale values were assigned to the templates arbitrarily as mentioned before, purely for the purpose of synthesizing an image. They were assigned to the templates independently of the assignment of grey-scale values in the panchromatic image and, hence, they could have a grey-scale coloring that differed greatly from that of the panchromatic image. Would they, nevertheless, represent a similar distribution of values among the pixels? Only a consistent mapping of grey-scale pixel value in the multispectral image to grey-scale pixel value in the panchromatic image was desired, not the sameness of the values themselves. The grey-scale values for the templates were assigned by first sorting the templates in decreasing order of the number of pixels with which they were associated (the number of 10-dimensional points in their clusters), and then assigning values starting with black and proceeding to white. The value white was assigned to all templates associated with a class containing 10 or fewer pixels (therefore, the value white differs from the others in being associated with more than one template).

As in Experiment 1, several combinations of values $\rho, t$ were tried to find the most
favorable combination for LimitsART 1 (with \( t > 0 \)) and the most favorable value of \( \rho \) for ART 1 (with \( t = 0 \)). Two evaluation criteria were applied to the results. The first was a visual inspection of the image quality obtained. Here, the goal was for the synthesized multispectral grey-scale image to match as closely as possible the panchromatic image as if the grey-scale pixel values of the latter had been permuted. The second evaluation criterion was a comparison of the average mutual information between the panchromatic image and the two multispectral images. This was calculated as the mutual information between two random variables \( X \) and \( Y \) via the equation

\[
I(X, Y) = \sum_{j=0}^{255} \sum_{k=1}^{K} p_{j,k} \log_2 \left( \frac{p_{j,k}}{p_j q_k} \right).
\]

Here, \( p_j \) is the probability that a pixel grey-scale intensity value \( j \) will occur in the panchromatic image, \( q_k \) is the probability that the template with an index value of \( k \) (where \( K \) templates have formed) will occur among the corresponding 160,000 multispectral pixel vectors. One may regard the index value \( k \) as a grey-scale value for a template, although in reality \( k \) is assigned a grey-scale value, which is then assigned to template \( k \). As has been stated, only the distribution of grey-scale values over image pixels in the multispectral and panchromatic images is important for our purpose, and not the values themselves. The probability \( p_{j,k} \) is the joint probability that a panchromatic pixel location has intensity value \( j \) and the corresponding multispectral pixel is assigned to the template with index \( k \). These numbers were computed from frequencies normalized by the number of pixels. An \( I(X, Y) \) value of 0 represents independence between the random variables, and greater values signify greater dependence or similarity.

9 Results

The panchromatic image is shown in Fig. 10. A “best” multispectral image (having greatest discernible resolution) occurred for ART 1 at a vigilance value of 0.795; it is shown in Fig. 11. The “best” result for LimitsART 1 occurred at any vigilance value in the range from zero to 0.55 and with a hyperbox maximum threshold \( t \) of 0.55; it is shown in Fig. 12. The ART 1 image was based upon 399 templates and the LimitsART 1 image was based upon 452 templates, essentially produced by \( F_{-1}^+ \)-enforced resets. The latter image is clearly of superior quality when compared to the panchromatic image. As it turned out, this could be determined by either visual inspection or by which \( I(X, Y) \) value was the greater.

With the exception of white, each grey-scale value in the bar legend in each of figs. 11 and 12 represents one template. Each value is labelled with a positive integer giving the number of pixels associated with that template. With the exception of white, decreasing grey-scale values are assigned to the templates in the order of decreasing density (number of pixels divided by number of hyperbox voxels) going from the bottom to the top of the bar. White represents all templates with 10 or fewer pixels in Fig. 11. To force the bar to have the same number of grey-scale values in Fig. 12 as in Fig. 11, we used a higher breaking point for the number of pixels for templates assigned the color white. Notice that the distribution of pixels over the templates falls off sharply in Fig. 12 by contrast with Fig. 11.

That LimitsART 1 yielded a superior image over that of ART 1 was established more rigorously by comparing \( I(X, Y) \) values obtained for each network as described in Section
Figure 11: Grey-scale (multispectral) image generated by ART 1, \( \rho = 0.795 \).
Figure 12: Grey-scale (multispectral) image generated by LimitsART 1, $\rho = 0.55, t = 0.55$, same number of templates (grey-scale values) as in Fig. 10.

7. The results are shown in Table 2. Notice that there is roughly twice the agreement between the LimitsART 1-generated image and the panchromatic image than between the ART 1-generated image and the panchromatic image.

10 Discussion and Conclusion

The objective of this paper was to present experiments testing the ability of the category-theoretic semantic theory introduced here to advance the state of the art in neural network design. Specific categorical constructs, colimits and limits, were applied to determine neural structures for the re-design of a neural network. The re-design improves the neural network’s

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Mutual Information $I(X, Y)$ in bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$</td>
<td>$t$</td>
</tr>
<tr>
<td>0.55</td>
<td>0.55</td>
</tr>
<tr>
<td>0.795</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 2: Information-theoretic comparison of each network in multispectral image construction versus panchromatic image.
ability to “learn” through connectionist adaptation a representation of the knowledge implicit in a set of multicomponent (multidimensional) input data patterns. This improvement stems from the exploitation of a knowledge representation capability which was missing in an ART network design. The missing capability was provided by adding to the network certain neural structures representing colimits and limits of diagrams of concepts and their morphisms. The redesigned network (LimitsART 1) formed connection-weight template patterns representing multidimensional hyperbox regions that apparently were a better classification scheme for the input data than the hyperbox regions provided by the original network.

One experiment tested the rationale underlying the category-theoretic modification in LimitsART 1. The criterion for success was a measure of the ability to form hyperbox templates which encompassed four well-separated, synthesized clusters. Another experiment, based upon synthetic satellite camera imagery, yielded a test of the ability of LimitsART 1 to outperform ART 1 on data from an application domain. Multispectral data for a low-resolution image with fewer pixels was used as multidimensional input in an ART 1 classification scenario and compared with a high-resolution panchromatic image as a measure of neural network performance. In both experiments LimitsART 1 apparently formed hyperbox templates leading to a better classification of the input patterns.

The hyperbox templates formed by LimitsART 1 were the result of the category-theoretic modification to apply an abstraction about hyperbox side length while processing the input patterns. LimitsART 1, unlike ART 1, was able thereby to control hyperbox shape by upper-bounding the hyperbox side length. The abstraction was the result of LimitsART 1’s representation of limits of concept diagrams involving colimits of concept diagrams. The concepts are represented in the network by the activations of specific network nodes while processing data, and the concept morphisms are represented by connection-weighted pathways between nodes.

The significance of these experimental results is that an abstraction derived from the semantic theory has given rise to a numerical engineering parameter. The abstraction is directly expressed by, indeed, was derived through the application of, category theory. Category theory is an abstract theory of structure, any sort of structure that can be represented mathematically, that is, systematically. Here, it has been shown to be applicable to neural network design, leading to an improved performance result in this experiment. This provides one piece of evidence for the general usefulness of the category-theoretic neural network semantic theory presented here. More research is needed to explore the possibilities of neural network design (or re-design) posed by the semantic theory. We hope that this result will encourage others to join in this research.

References


