# From Categorical Semantics to Neural Network Design

Michael J. Healy
University of New Mexico and
University of Washington
13544 23rd Place NE
Seattle, WA 98125, USA
mjhealy@u.washington.edu

Thomas P. Caudell, Yunhai Xiao
University of New Mexico
Department of Electrical and
Computer Engineering
Albuquerque, New Mexico 87131, USA
tpc,yhxiao@eece.unm.edu

**Abstract - We introduce a new architecture designed by applying a recently-developed mathematical model of neural network semantics using category theory. The new design has multiple subnetworks associated with different sensors and association regions. The subnetworks form individual, hierarchical representations of a body of knowledge. Subnetwork interconnections adapt to link the individual concept representations appropriately and provide knowledge coherence, representing a single knowledge heirarchy across the multi-sensor network.**

## I INTRODUCTION

Some researchers have been investigating the capabilities of neural networks in terms of their ability to acquire and use knowledge [1, 2, 3, 4, 5]. We have been developing a mathematical semantic model to achieve a high level of rigor in analyzing the acquisition and representation of knowledge by neural networks[6, 7, 8]. Category theory is the principal mathematical discipline applied in the model. Here, we describe its application to the design of a neural network architecture for knowledge acquisition from input data examples processed through two or more sensors. We begin with an introduction to categorical semantic modeling, then discuss the modeling specific to the architecture's design. The design serves as an example application of the semantic model and provides an experimental device for further research.

## II CATEGORICAL SEMANTICS

There are many introductions to category theory, examples being[9] and [6]. A category has objects, which are examples of some type of mathematically-definable structure, and morphisms, which are relationships between pairs of objects that express some aspect of that type of structure, together with a law of composition by which properly-aligned pairs of morphisms can be combined to calculate other morphisms. A morphism directed from object $a$ (its domain) to object $b$ (its codomain) is denoted $f: a \longrightarrow b$. The composition for an aligned pair $f : a \longrightarrow b$ and $g : b \longrightarrow c$ (meeting head to tail at an object $b$) in a category is a morphism from $a$ to $c$, written $g \circ f : a \longrightarrow c$. Composition is associative, $h \circ (g \circ f) = (h \circ g) \circ f$, and each object $a$ has an *identity morphism* $\mathtt{id}_a: a \longrightarrow a$ such that for any $f: a \longrightarrow b$ and $g: c \longrightarrow a$, $\mathtt{id}_a \circ g = g$ and $f \circ \mathtt{id}_a = f$. The ability to form compositions of morphisms makes it possible to formulate and solve equations involving them.

A diagram in a category $\mathcal{C}$ appears as a figure containing some of the objects and morphisms of $\mathcal{C}$. A commutative (or commuting) diagram is one with the property that for any two paths through the diagram morphisms between a pair of diagram objects $a$ and $b$, if at least one of the paths contains two or more morphisms, then the composition of the morphisms along either path yields one and the same result morphism. A colimit for a diagram is an object (the colimit object) and a collection of morphisms that extends the diagram to a commutative diagram with minimal added information[6]. Colimits are used extensively to represent the learning of new concepts by a neural network based upon the merging of pre-existing concepts, beginning with the simple concepts about stimuli represented by the input nodes. Here, we shall restrict colimits to the simplest kind, coproducts, which are the colimits of discrete diagrams (diagrams consisting only of objects, with no morphisms). As an example, consider a discrete diagram containing two objects $a$ and $b$. A coproduct for this diagram consists of a coproduct object $a + b$ and two morphisms $i_1 : a \longrightarrow a + b$ and $i_2 : b \longrightarrow a + b$ called the injections of the coproduct. A coproduct object for $a$ and $b$ in the category of sets and functions is a disjoint union, the coproduct morphisms being the injection functions of $a$ and $b$ separately into $a + b$.

Structure-preserving mappings between categories, called functors, are basic to category theory. A functor $\mathbf{F} : \mathcal{C} \longrightarrow \mathcal{D}$ from a category $\mathcal{C}$ to a category $\mathcal{D}$ maps the objects $a, b$ and morphisms $f : a \longrightarrow b$ of $\mathcal{C}$ to unique images $\mathbf{F}(a)$, $\mathbf{F}(b)$ and $\mathbf{F}(f) : \mathbf{F}(a) \longrightarrow \mathbf{F}(b)$ in $\mathcal{D}$. A functor $\mathbf{F}$ is structure-preserving by virtue of two properties it has with respect to the compositions in $\mathcal{C}$

0-7803-7898-9/03/$17.00 ©2003 IEEE

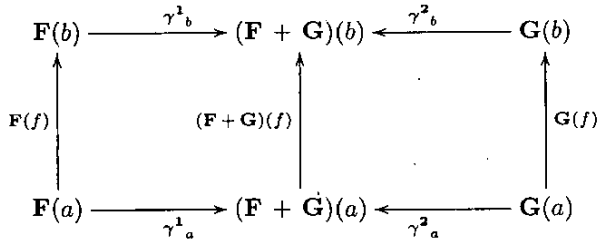



Figure 1: **A coproduct in $\mathcal{D}^{\mathcal{C}}$ "pastes together" commutative squares along the morphism images of the coproduct functor.**

and $\mathcal{D}$: $\mathbf{F}(g \circ_{\mathcal{C}} f) = \mathbf{F}(g) \circ_{\mathcal{D}} \mathbf{F}(f)$ (where $g$ is a morphism with domain $b$, so $g \circ_{\mathcal{C}} f$ is defined in $\mathcal{C}$) and $\mathbf{F}(\mathrm{id}_a) = \mathrm{id}_{\mathbf{F}(a)}$. Functors can be many-to-one on either or both objects and morphisms; for two objects $a$ and $b$, for example, it can be that $\mathbf{F}(a) = \mathbf{F}(b)$. Because of its significance in architecture design[7, 8], we refer to this "merging of objects and morphisms" as *compression.*

A natural transformation $\gamma : \mathbf{F} \longrightarrow \mathbf{G}$ between functors $\mathbf{F}, \mathbf{G} : \mathcal{C} \longrightarrow \mathcal{D}$ consists of $\mathcal{D}$-morphisms $\gamma_a$, one for each object $a$ of $\mathcal{C}$, such that for each morphism $f: a \longrightarrow b$ of $\mathcal{C}$, $\mathbf{G}(f) \circ \gamma_a = \gamma_b \circ \mathbf{F}(f)$. The square-shaped commutative diagrams for two natural transformations $\gamma^1, \gamma^2 : \mathbf{F}, \mathbf{G} \longrightarrow (\mathbf{F} + \mathbf{G})$ evaluated at two objects $a$ and $b$ and a morphism $f: a \longrightarrow b$ are shown in Fig. 1. This figure also illustrates the notion of a coproduct for two functors $\mathbf{F}$ and $\mathbf{G}$. Given categories $\mathcal{C}$ and $\mathcal{D}$, there is a category $\mathcal{D}^{\mathcal{C}}$ whose objects are the functors $\mathbf{F} : \mathcal{C} \longrightarrow \mathcal{D}$ and whose morphisms are the natural transformations, such as $\gamma^1$, $\gamma^2$. A coproduct functor $\mathbf{F} + \mathbf{G}$ for two functors $\mathbf{F}$, $\mathbf{G}$ in $\mathcal{D}^{\mathcal{C}}$ (provided the required injection morphisms exist) is one that compresses a pair of objects or of morphisms from $\mathcal{C}$ if and only if the pair is compressed by both $\mathbf{F}$ and $\mathbf{G}$. The coproduct has the effect of "pasting together" the corresponding commutative squares of the natural transformation injections as shown in Fig. 1.

The semantic model begins with a category we call **Concept**; its objects are concepts (expressed as theories in a formal logic) and its morphisms are relations that show how one concept is used in another (actually theory morphisms, which map the sorts and formulas in one theory into another so that the algebraic strcuture of logic is preserved and axioms map to axioms or theorems). This is our mathematical representation of the hierarchical structure of knowledge, and is meant include everything a neural network will ever learn.

For each neural network architecture, we define another category, which we shall call **Neural** here. The objects of **Neural** are intervals on the real line, serving as the quanta in a quantized representation of the output levels of the nodes. In the initial categorically-derived architecture presented here, the nodes can be considered as binary. Hence, for each node $p_i\,(0 \leq i \leq n)$, where there are $n$ nodes, we need consider only two objects, $(p_i, \eta_{i,1})$, $(p_i, \eta_{i,2})$, where $\eta_{i,1} = \{x \mid -\infty < x \leq 0\}$, $\eta_{i,2} = \{x \mid 0 < x < +\infty\}$. For brevity, we omit a detailed description of neural morphisms, which are somewhat complex. Briefly, a morphism from $(p_i, \eta)$ to $(p_j, \eta')$ (where $\eta, \eta'$ are arbitrary intervals) consists of (1) a set of weights for the network and (2) a bundle of connection paths (each path having possibly more than one intermediate node, i.e., multiple connections) satisfying the following conditions: (a) the paths all originate at $p_i$ and terminate at $p_j$; (b) they are all activated by the same set of network inputs for any weight vector in the set so that $p_i, p_j$ emit outputs in the intervals $\eta, \eta'$, respectively, and all intermediate path nodes of the paths in the bundle emit outputs in specified intervals (see[6]; for simplicity, we have substituted inputs for node activation state sequences).

An algorithm for the colimit construction in **Concept** [6] enables the calculation of complex concepts via the colimits of diagrams involving simpler concepts and morphisms. Functors in the category $\mathbf{Neural}^{\mathbf{Concept}}$ preserve the colimit construction, in part because functors preserve diagram commutativity. A given functor $\mathbf{M}_A$: **Concept** $\longrightarrow$ **Neural**, therefore, effectively transports the structure of all possible knowledge— represented by the concept category—into the categorical architecture representation. The colimit construction also indicates how concepts are formed through adaptation. Of necessity, $\mathbf{M}_A$ entails a large amount of compression; in fact, it represents a single state of learning in the network, with all other knowledge compressed. Many such functors are required to represent many stages of learning.

Of particular interest to us are networks having several subnetwork regions, each providing network computations associated with a sensor or an actuator or performing some other function, such as serving as an association region for two or more sensor regions. Correspondingly, we use several functors from $\mathbf{Neural}^{\mathbf{Concept}}$ to represent a given state of learning, where each functor models the existing knowledge representation within a region. Our goal is to blend the individual, regional knowledge representations to achieve a single working knowledge representation. We call this knowledge coherence, which we model as a system of natural transformations connecting the region functors.

To see how we apply this in analyzing existing neural network designs, consider some of the ART mapping networks (for example, as analysed in a rule-base analysis in [3]). These couple separate, unsupervised ART networks (sometimes with an intermediate "mapping field" subnetwork) to provide supervised ART systems. We have

shown[7] that each ART network has solely coproduct formation as a means of learning complex concepts in terms of the simple concepts represented by its input nodes. This stems from the lack of connections within the input ( $F_0$ ) and matching ( $F_1$ ) layers. The more general colimits required to capture knowledge representation are missing. Also, there are no nodes available for compression of yet more complex (or less complex) concepts, or input concepts not represented by the present input layer. The effect is to limit the concept hierarchy representation to a functor image fragment containing at most coproducts over a single layer. Further, the interconnections between ART networks required for commutative squares are not present, so any natural transformations would be only partially represented[7].

On the positive side, ART networks have important advantages for a study of knowledge representation. The explicit learning procedure and the presence of feedback (top-down) connections facilitates the identification of intervals $\eta$ of the $F_2$ nodes as coproduct objects: The bottom-up connections to a previously-committed $F_2$ node that have nonzero weights are the connection paths of the coproduct injection morphisms. Using an ART 1 network as an example[10], the coproduct objects are the previously-mentioned intervals $\eta_2$ of the committed $F_2$ nodes. Suppose that the template of a winner-take-all node in the $F_2$ competition to represent an input pattern is a subset template of "sufficient size" (see [10]; "subset" means that each nonzero template connection weight corresponds to a nonzero input pattern value, and "sufficient size" means that there are "enough" template nonzeros). The top-down, unit-weight template connections then ensure the activation in their $\eta_2$ intervals of exactly the $F_1$ nodes in the discrete diagram upon which the currently-expressed coproduct is based, thus ensuring the continued activation of the coproduct morphism connections. This largely describes the semantics of resonance in a typical ART network.

The interconnects between $F_2$ layers of the separate ART networks is suggestive of natural transformation components corresponding to the concepts represented by the $F_2$ $\eta_2$ intervals. What is missing are the components corresponding to the concepts at the $F_0$ (or, equivalently, the $F_1$ ) level. We decided to exploit the properties of network templates and interconnections by using ART networks as a point of departure in developing new architectures more compatible with knowledge representation and coherence.

## III A CATEGORICALLY-MOTIVATED DESIGN

We have given a very brief example of applying the categorical semantic theory using ART mapping networks. The example emphasized binary nodes for simplicity, using the binary intervals $\eta_2$. Our other example, the proposed categorical neural architecture that is the subject of this paper, will be described next. Although this initial architecture is designed to operate in a binary fashion, We use the more general notation $\eta$ for arbitrary intervals for simplicity and also to emphasize that the semantic model applies to other than binary-node networks.

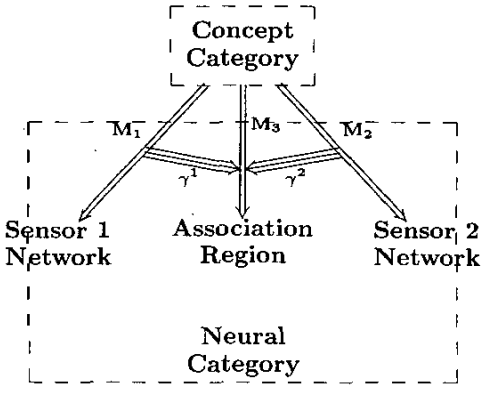



Figure 2: **Functors map the hierarchy of a concept category to multiple regions. Natural transformations represent coherent interconnections between hierarchy representations.**

Fig. 2 shows the overall scheme for an architecture that represents and coherently fuses the information from two sensors according to our theory. There are three regions, two receiving input from sensors and the third, an association region, receiving input from the other two regions. In terms of the semantic model, the architecture is represented by the category **Neural** and the sensor regions are represented by the images of the functors $\mathbf{M}_1$ and $\mathbf{M}_2$. The association region is represented by the image of $\mathbf{M}_3$. Two natural transformations $\gamma^i : \mathbf{M}_i \longrightarrow \mathbf{M}_3$ ($i \in \{1, 2\}$) are coproduct morphisms for the discrete diagram containing $\mathbf{M}_1$, $\mathbf{M}_2$ in $\mathbf{Neural}^{\mathbf{Concept}}$, and $\mathbf{M}_3$ is the corresponding colimit object, $\mathbf{M}_3 = \mathbf{M}_1 + \mathbf{M}_2$. Fig. 3 shows an initial architecture designed in accordance with this scheme and using properties of ART networks where these can be exploited. We call the sensor regions *primaries* and denote these by $\mathbf{P}_1$ and $\mathbf{P}_2$, corresponding to the functors $\mathbf{M}_1$ and $\mathbf{M}_2$. We call the region corresponding to $\mathbf{M}_3$ an *associator*, which we will denote by $\mathbf{A}$. Adaptive connections are indicated by arrows ending in bullets and non-adaptive connections have standard arrowheads. Connection polarity is indicated by + (excitatory), - (inhibitory), or $R$ (a bundle of non-phasic reset connections).

An immediate problem occurs in implementing the scheme of Fig. 2 as a multi-sensor architecture. Let us suppose that the two sensors are logically independent. For example, let $\mathbf{P}_1$ be coupled to a visual sensor, with its $F_0$ layer representing concepts describing visual prim-

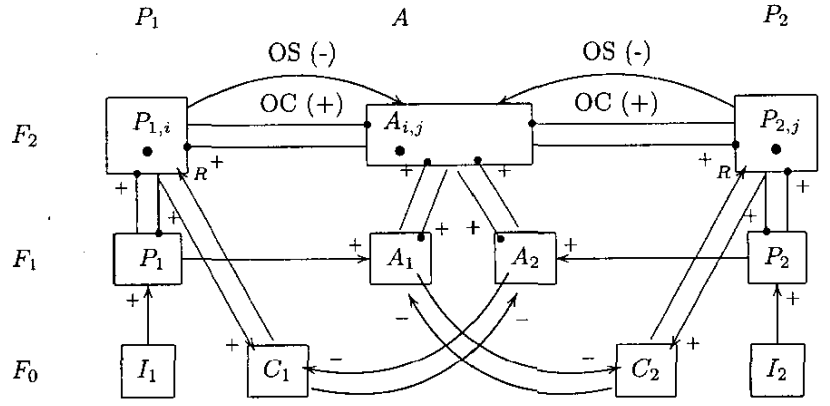



Figure 3: **A schematic for NR 1.3, an architecture design based upon three objects (functors) $P_1$, $A$ and $P_2$ and two morphisms (natural transformations) $\gamma^1: P_1 \longrightarrow A$ and $\gamma^2: P_2 \longrightarrow A$ in the functor category Neural$^{\text{Concept}}$.**

itives of some kind, and let the $F_0$ nodes at $\mathbf{P}_2$ represent tactile information provided by an array of pressure sensors. Then, since the two kinds of sensor primitives convey entirely different kinds of infornmation, each subnetwork must omit the input concepts associated with the other subnetwork's sensor. However, our semantic model requires that each functor map the entire knowledge space of **Concept** into its subnetwork region, including all of the concepts that describe both kinds of sensor primitives. To satisfy these competing requirements, there must be **Neural** objects and morphisms available to represent the functorial compression of sensor primitives for each subnetwork within the other subnetwork. We choose to represent these as a sort of auxiliary $F_1$ node, because it is at $F_1$ that the discrete diagrams form via bottom-up/top-down matching (ART-based, although the gain control subsystem is not shown). We include one of these compression nodes in each subnetwork to serve as the target for the functor images of the missing sensor primitives. They are labelled $C_1$ (for $\mathbf{P}_1$) and $C_2$ (for $\mathbf{P}_2$) in Fig. 3. The coproduct functor $\mathbf{M}_3$, on the other hand, fully represents the non-compressed knowledge represented by $\mathbf{M}_1$ and $\mathbf{M}_2$. Therefore, region $\mathbf{A}$ has two $F_1$ layers whose nodes receive inputs from their corresponding $F_1$ nodes in $\mathbf{P}_1$ and $\mathbf{P}_2$, respectively. We use the term "proxies" for the two $F_1$ layers in $\mathbf{A}$; however, their bottom-up/top-down connections with the $\mathbf{A}$ $F_2$ layer operate independently of $\mathbf{P}_1$ and $\mathbf{P}_2$.

We connect each compression node to the proxy of the opposite subnetwork's $F_1$ layer as shown. There are pairs of connection paths emanating from the $F_1$ nodes of $\mathbf{P}_1$ and $\mathbf{P}_2$ to the appropriate $\mathbf{A}$ $F_2$ nodes. These form the commutative squares of the two natural transformations $\gamma^1: \mathbf{M}_1 \longrightarrow \mathbf{M}_3$ and $\gamma^2: \mathbf{M}_2 \longrightarrow \mathbf{M}_3$. For example, let $S$ be an object in **Concept** that descibes a sensor primitive via the functor $\mathbf{M}_1$, so that for some $\mathbf{P}_1$ $F_1$ node $F^1_{1,k}$ and an appropriate interval $\eta$, $(F^1_{1,k}, \eta) = \mathbf{M}_1(S)$. To simplify notation, we shall simply use the functorial representation, in the present case $\mathbf{M}_1(S)$. Proceeding, let $S$ be represented in the $\mathbf{A}$ subnetwork's $\mathbf{P}_1$ $F_1$-proxy layer via the functor $\mathbf{M}_3$ as $\mathbf{M}_3(S)$. This yields two $F_1$ representations of $S$, connected across subnetworks by a $\mathbf{P}_1$ $F_1$ to $\mathbf{A}$ $F_1$ connection that represents the natural transformation component $\gamma^1(S): \mathbf{M}_1(S) \longrightarrow \mathbf{M}_3(S)$. Let $T$ be a coproduct concept with a coproduct injection morphism $m: S \longrightarrow T$, and let $T$ be represented in the $\mathbf{P}_1$ and $\mathbf{A}$ $F_2$ layers by $\mathbf{M}_1(T)$ and $\mathbf{M}_3(T)$, respectively. Then, there are two connection paths from $\mathbf{M}_1(S)$ to $\mathbf{M}_3(T)$ forming the sides of a commutative square, as follows. One path consists of two connections: a $\mathbf{P}_1$ $F_1$ to $\mathbf{P}_1$ $F_2$ connection, the coproduct morphism image $\mathbf{M}_1(m): \mathbf{M}_1(S) \longrightarrow \mathbf{M}_1(T)$, followed by the $\mathbf{P}_1$ $F_2$ to $\mathbf{A}$ $F_2$ connection for the $\gamma^1$ component $\gamma^1(T): \mathbf{M}_1(T) \longrightarrow \mathbf{M}_3(T)$, yielding the composition morphism $\gamma^1(T) \circ \mathbf{M}_1(m): \mathbf{M}_1(S) \longrightarrow \mathbf{M}_3(T)$. The second path consists of two connections: a $\mathbf{P}_1$ $F_1$ to $\mathbf{A}$ $F_1$ connection for the $\gamma^1$ component $\gamma^1(S): \mathbf{M}_1(S) \longrightarrow \mathbf{M}_3(S)$, followed by an $\mathbf{A}$ $F_1$ to $\mathbf{A}$ $F_2$ connection, the coproduct morphism image $\mathbf{M}_3(m): \mathbf{M}_3(S) \longrightarrow \mathbf{M}_3(T)$, yielding the composition morphism $\mathbf{M}_3(m) \circ \gamma^1(S): \mathbf{M}_1(S) \longrightarrow \mathbf{M}_3(T)$. The composition morphisms are the same morphism, $\gamma^1(T) \circ \mathbf{M}_1(m) = \mathbf{M}_3(m) \circ \gamma^1(S)$, hence, in the architecture, the two connection paths must represent the same **Neural** morphism. In other words, the square whose sides are the four factors in the two compositions is commutative. Thus, the relationship between the primary sensor primitive concept representation $\mathbf{M}_1(S)$ and the associator colimit concept representation $\mathbf{M}_3(T)$, where concept $T$ includes concept $S$ as one of its parts, is independent of the path that expresses the relationship across the primary $\mathbf{P}_1$ and the associator $\mathbf{A}$. This is the basis for knowledge coherence and is one of the theoretical con-

siderations that help determine the operational rules for the architecture.

Joined to the preceding commutative square is one for the natural transformation $\gamma^2: \mathbf{M}_2 \longrightarrow \mathbf{M}_3$. In the latter, we obtain two connection paths from $\mathbf{M}_2(S)$ to $\mathbf{M}_3(T)$; however, there is no $\mathbf{P}_2$ $F_1$ node representing $S$, but instead the compression node $C_2$. Defining the appropriate functor images and natural transformation components as before, but this time using $C_2$ instead of a $\mathbf{P}_2$ $F_1$ node (but using again the same $\mathbf{A}$ $F_1$ node from the proxy layer for $\mathbf{P}_1$ $F_1$), we obtain the commutative square associated with the equality $\gamma^2(T) \circ \mathbf{M}_2(m) = \mathbf{M}_3(m) \circ \gamma^2(S)$. The fact that the two commutative squares are "pasted together" along $\mathbf{M}_3(m): \mathbf{M}_3(S) \longrightarrow \mathbf{M}_3(T)$ implies that in the operational architecture, all four connection paths must become activated for the same inputs. The details of operation must be defined to be consistent with this.

## IV DISCUSSION OF OPERATION

There is not space here to describe the operational flow of the new architecture in full, but we can point out a few guiding principles. First, the fact that the architecture is to implement coproducts, functors and natural transformations and, in particular, coproducts in $\mathbf{Neural}^{\mathbf{Concept}}$, serves as a basis for the algorithm for the activation and connection-weight modifications. Thus, the appropriate commutative square diagrams must be activated connecting the $\mathbf{P}_1$ and $\mathbf{P}_2$ $F_1$ and compression nodes to the currently-active $\mathbf{A}$ $F_2$ coproduct node. This operational consideration is a realization of a theoretical implication of the semantic model: The fusion of information from multiple sensors is a consequence of knowledge coherence, fully expressed as the calculation of colimits at two levels. First, colimits in **Concept** are mapped to their neural representations by functors, which preserve the commutativity of diagrams. Second, natural transformations representing interconnects to an association region in a neural network define the latter as a coproduct of functors; in the presented architecture, this results in the appropriate "pasting together" of the commutative squares some of whose sides are the colimit (in this case, coproduct) morphism images for the functors.

More operational detail is provided by the desire to stay as close as possible to ART network design. The $F_2$ choice and template modification operations within each subnetwork proceed according to the ART gain control and connectionist adaptation mechanisms. For example, **Neural** morphisms such as $\mathbf{M}_1(m): \mathbf{M}_1(S) \longrightarrow \mathbf{M}_1(T)$ and $\mathbf{M}_3(m): \mathbf{M}_3(S) \longrightarrow \mathbf{M}_3(T)$ in the illustration of commutative squares for $\gamma^1: \mathbf{M}_1 \longrightarrow \mathbf{M}_3$ are implemented as bottom-up connections, but the latter are assisted in this role by their corresponding top-down template connections, which ensure their activation during the appropriate periods.

The principle of connections not directly involved in a morphism supporting those that are in their role is extended throughout the new architecture. For example, the morphism $\gamma^1(T): \mathbf{M}_1(T) \longrightarrow \mathbf{M}_3(T)$ is implemented as a $\mathbf{P}_1$-$F_2$-to-$\mathbf{A}$-$F_2$ connection (an on-center (OC) connection in Fig. 3), but the latter receives support in that role from its attendant $F_2$-to-$F_2$ off-surround (OS) connections to the rest of the $\mathbf{A}$ $F_2$ layer and also from a reciprocal connection with which it is paired. Thus, coproduct nodes in $\mathbf{P}_i$ and $\mathbf{A}$ $(i = 1, 2)$ that represent the same coproduct concept form mutually-supportive pairs. Sustained activation of a chosen $\mathbf{P}_1$ $F_2$ node, for example, depends upon this, for its support from its own template weakens during the time the input it is representing is presented to the network.

The $\mathbf{P}_2$-to-$\mathbf{A}$ commutative square that shares a side with the previously-described $\mathbf{P}_1$-$F_1$-to-$\mathbf{A}$-$F_2$ square has the $\mathbf{P}_2$ compression node $C_2$ in the role of the third representation of concept $S$, $\mathbf{M}_2(S)$, and its connections to/from the $\mathbf{P}_1$ $F_1$ proxy in $\mathbf{A}$ are inhibitory. To examine the effect this has in a typical operational scenario, let us assume that for the current episode of presentation of input patterns to the $\mathbf{P}_1$ and $\mathbf{P}_2$ $F_0$ layers, $\mathbf{P}_1$ was the first to reach its candidate for its $F_2$ choice, a node representing the functor image $\mathbf{M}_1(T)$ of a concept $T$. Suppose that, next, a node representing the image $\mathbf{M}_2(T')$ of a concept $T'$ becomes activated. This causes the activation of $C_2$ (see the appropriate top-down connection in Fig. 3), which then acts to suppress the $\mathbf{P}_1$ $F_1$ proxy layer in $\mathbf{A}$; but $\mathbf{M}_1(T)$ has already provided excitation to $C_1$, which acts in a similar manner to suppress the $\mathbf{P}_2$ $F_1$ proxy. Under an ART-like "2/3 Rule" operating in $\mathbf{A}$, the proxies can only sustain their activities if the nodes representing $\mathbf{M}_3(T')$ and $\mathbf{M}_3(T)$ can provide the top-down excitatory stimulation through the proxy template connections. However, if $\mathbf{M}_3(T') \neq \mathbf{M}_3(T)$, then two distinct $\mathbf{A}$ $F_2$ nodes will be competing for activation as winner-take-all nodes. This results in their mutual suppression through the OS inhibitory connections from $\mathbf{P}_1$ and $\mathbf{P}_2$, resulting in loss of top-down support to their proxies. The final result is a loss of activity in the node representing $\mathbf{M}_1(T)$, for its support from its template has weakened with the passage of time, as mentioned before (in the current architectural design, $\mathbf{M}_2(T)$, having reached its choice later, has support from its "fresh" nonzero template bottom-up connections, hence, remains near full activation). The weakened state of the $\mathbf{M}_1(T)$ node allows the reset connection from $C_1$ to the $\mathbf{P}_1$ $F_2$ layer to have its full effect ($C_1$, like $\mathbf{P}_2$, lags behind $\mathbf{P}_1$ $F_1$ and $F_2$ in its cycle of activation). This results in the stimulation of the $F_2$ competitors for $\mathbf{M}_1(T)$, which then compete to choose a new winner. Thus, the functor co-

product is enforced by rejecting the choice of the primary subnetwork that made the original prediction.

If, on the other hand, $\mathbf{M}_3(T') = \mathbf{M}_3(T)$, then the primary subnetwork predictions are consistent and the entire network is said to be in resonance. The proper associations between concept representatives have been established and enforced at both the $F_1$ and $F_2$ levels across all three subnetworks. Thus, resonance as here described corresponds to coherence at both concept levels. That is, a concept representation at $F_1$ in both primaries is related to a single associator $F_2$ representation through a morphism representation, a bundle of connections representing a commutative square. The two commutative squares express the notion that the primary-to-associator subconcept-to-concept relationships are independent of path, hence, are expressed without ambiguity.

This architecture is currently in preliminary testing for a series of experiments that will follow. Experience gained with it so far has led to some design refinements. The refinements are not completely determined by expediency: When a difficulty arises, it leads us to consider the semantic model, and to ask if we are being fully consistent with it and to what extent does it determine the architectural details.

## V CONCLUSION

We have derived novel architecture design principles from a semantic model for neural networks based upon category theory. We have given two examples to illustrate the use of this model: Its application to understanding the semantics of ART networks, and the design of a multi-region, multi-sensor neural network capable of coherent knowledge representation based upon the new design principles. In the new design, nodes and connections are explicitly organized to implement coproducts, functors and natural transformations. Experimentation with the new architecture is under way.

There are still details implied by the semantic model that are missing in the new architecture introduced here. First, as in ART, the colimits represented in each subnetwork are only two-layer coproducts. Second, the compression nodes and morphisms for the infinite variety of concepts not representable in any of the three subnetworks are missing, with the sole exception of the compression nodes for the alternate sensor in each sensor subnetwork. Nevertheless, the architecture described here has filled in some of the semantic detail. It is a start at fully realizing the semantic model in architectural design.

# References


[1] R. Andrews, J. Diederich and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks", *Knowledge-Based Systems*, vol. 8, no. 6, 373-389, 1995.

[2] M. W. Craven and J. W. Shavlik, "Learning symbolic rules using artificial neural networks", *Proceedings of the 10th International Machine Learning Conference*, Amherst, MA. San Mateo, CA:Morgan Kaufmann, 1993, pp. 73-80.

[3] M. J. Healy and T. P. Caudell, "Acquiring rule sets as a product of learning in a logical neural architecture", *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 461-475, 1997.

[4] N. K. Kasabov, "Adaptable neuro production systems", *Neurocomputing*, vol. 13, pp. 95-117, 1996.

[5] G. Pinkas, "Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge", *Artificial Intelligence* vol. 77, pp. 203-247, 1995.

[6] M. J. Healy, "Category theory applied to neural modeling and graphical representations", in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, July 24-27, Como, Italy, IEEE CS Press, vol. III, 2000, pp. 35-40.

[7] M. J. Healy and T. P. Caudell, "A categorical semantic analysis of ART architectures", *IJCNN'01:International Joint Conference on Neural Networks*, Washington, DC. IEEE Press. vol. 1, 2001, pp. 38-43.

[8] M. J. Healy and T. P. Caudell, "Aphasic compressed representations: a functorial semantic design principle for coupled ART networks", *IJCNN'02:International Joint Conference on Neural Networks*, Honolulu, Hawaii. CD-ROM Proceedings. The Printing House, Inc.:Stoughton, WI, 2002, P. 2656.

[9] B. C. Pierce, *Basic Category Theory for Computer Scientists*, MIT Press, 1991.

[10] G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine", *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 54-115, 1987.