Category Theory Applied to Neural Modeling and Graphical Representations

Michael J. Healy mjhealy@u.washington.edu
13544 23rd Place NE
Seattle, WA 98125

Abstract

Category theory can be applied to mathematically model the semantics of cognitive neural systems. Here, we employ colimits, functors and natural transformations to model the implementation of concept hierarchies in neural networks equipped with multiple sensors.

1 Introduction

In this paper, we describe a mathematical scheme for the analysis and design of cognitive neural network architectures based upon functors and natural transformations, the structural mappings of category theory. In a previous paper[3], we described a mathematical scheme for representing the hierarchical structure of subconcept-concept relationships based upon colimits, a categorical construction for objects which represent entire diagrams, or structural graphs, of related objects. Functors map the concept colimits into a category of neural components; natural transformations between the functors unify single-sensor concept representations in a fused, multi-mode neural network architecture. This kind of mathematical modeling addresses the semantics behind the organization of neural systems.

In our scheme, connection-weight patterns representing complex concepts form when a neural network coupled to one or more sensors is stimulated by events that call up simpler component concept representations. A description of a concept with full mathematical rigor can be regarded as a theory in a system of theories written in formal logic[2]. Each theory is a logical description of some "domain" of physical or imagined quantities that behave in a coherent manner. It has a finite description in terms of axioms, the kinds of quantities it represents—called types or sorts—and mathematical operations and relations defined upon the sorts. For our purposes, theories describe items experienced by a neural network through its sensor inputs.

We assume that a node becomes activated when it receives enough input through some subset of its excitatory input connections to overcome any inputs that it receives through inhibitory input connections. Once activated, it sends an output signal ξ to other nodes or to an output device; its output is generated by a signal function ϕ, where $\phi(\theta - \theta_\kappa) > 0$ if $\theta - \theta_\kappa > 0$ and $\phi(\theta - \theta_\kappa) = 0$ if $\theta - \theta_\kappa \leq 0$. Here, $\theta = \Sigma_{\mu \in I^+} w_\mu^+ \delta_\mu - \Sigma_{\mu \in I^-} w_\mu^- \delta_\mu$, where $w_\mu^+ \quad (\mu \in I^+)$ and $w_\mu^- \quad (\mu \in I^-)$ are the excitatory and inhibitory connection weights for the input connections, with $w_\mu^+, w_\mu^- > 0$, and $\delta_\mu \quad (\mu \in I^+)$ and $\delta_\mu \quad (\mu \in I^-)$ are the current input signals along the respective excitatory and inhibitory connections. The quantity θ is the activation potential of the node and θ_κ, with $\theta_\kappa \geq 0$, is its activation threshold.

2 Category theory

A straightforward introduction to category theory is contained in [4]. The primitive quantities in category theory are *objects* and *morphisms*. Each morphism $f: a \longrightarrow b$ has a *domain* object a and a *codomain* object b. In a category, each pair of arrows $f : a \longrightarrow b$ and $g : b \longrightarrow c$ with a head-to-tail match (where the codomain b of f is also the domain of g) has a *composition* arrow $g \circ f : a \longrightarrow c$ whose domain a is the domain of f and whose codomain c is the codomain of g. Composition satisfies the associative law:

Diagram D Diagram D'

Figure 1: **Bottom: Two diagrams in the category $\mathbf{Nat}_{|}$.**

In triples which have a head-to-tail match by pairs, $f\colon a \longrightarrow b$, $g\colon b \longrightarrow c$ and $h\colon c \longrightarrow d$, the result of composition is order-independent, $h \circ (g \circ f) = (h \circ g) \circ f$. Also, for each object a, there is an *identity morphism* $\mathtt{id}_a\colon a \longrightarrow a$ such that for any arrows $f\colon a \longrightarrow b$ and $g\colon c \longrightarrow a$ $\mathtt{id}_a \circ g = g$ and $f \circ \mathtt{id}_a = f$.

There is a category **Set** whose objects are sets, whose morphisms are functions, and for which composition is just the familiar composition of functions, with $(g \circ f)(x) = g(f(x))$ for functions $f\colon a \longrightarrow b$ and $g\colon b \longrightarrow c$, with $x \in a$ and $(g \circ f)(x) \in c$. (Recall that function composition is associative and there is always an identity function that behaves as specified). To dispel any notion that the morphisms in a category always represent mappings, there is a category $\mathbf{N}_{|}^{+}$, in which the objects are nonzero natural numbers and in which there is a morphism $|_{n,m}\colon n \longrightarrow m$ if n is a divisor of m, denoted $n \mid m$. Notice that transitivity of the divisor relation yields associativity and that, since every nonzero natural number divides itself, identities exist.

A key concept in category theory is that of a commutative diagram—the categorical equivalent of a system of equations, but more general in nature. Diagram D' in Figure 1 is an example. The other diagram, D, can be seen as a formalization of the knowledge that 4 and 6 are both divisible by 2. Diagram D' formalizes this and the additional knowledge that 24 is divisible by 4 and 6. The dashed arrow represents the consequential knowledge that (because 24 is divisible by 4 and 6 while 4 and 6 are divisible by 2) 24 is divisible by 2. Notice that there are two morphisms from 2 to 24 that are compositions along a path directed through a third object (4 and 6, respectively), yet there is at most one divisibility morphism from one natural number to another. Therefore, $|_{4,24} \circ |_{2,4} = |_{2,24} = |_{6,24} \circ |_{2,6}$. The diagram D' is said to be a commutative diagram, one in which any two morphisms between a given pair of objects in the diagram are equal, provided that at least one of them is the composition along a path through one or more intermediate objects. Commutative diagrams are of great importance in category theory, for they serve to specify the constraints on structures. One of the most important uses of commutative diagrams is in defining colimits, which we shall encounter in a category of concepts.

A functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ from category $\mathcal{C}$ to category $\mathcal{D}$ maps the structure of $\mathcal{C}$ into that of $\mathcal{D}$. It associates each object a in $\mathcal{C}$ with a unique image object $F(a)$ in $\mathcal{D}$ and each morphism $f : a \longrightarrow b$ in $\mathcal{C}$ with a unique morphism $F(f) : F(a) \longrightarrow F(b)$ in $\mathcal{D}$. In addition, it preserves the compositional structure of $\mathcal{C}$ through the equations $F(g \circ_{\mathcal{C}} f) = F(g) \circ_{\mathcal{D}} F(f)$ (where $g \circ_{\mathcal{C}} f$ is defined in $\mathcal{C}$) and $F(\mathtt{id}_a) = \mathtt{id}_{F(a)}$ for all objects a of $\mathcal{C}$.

A natural transformation $\gamma : F \longrightarrow G$ between functors $F, G : \mathcal{C} \longrightarrow \mathcal{D}$ consists of $\mathcal{D}$-morphisms γ_a, one for each object a of $\mathcal{C}$, such that the following diagram in $\mathcal{D}$ commutes for each a:

$$\begin{array}{ccc} F(a) & \xrightarrow{\gamma_a} & G(a) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(f)} \\ F(b) & \xrightarrow[\gamma_b]{} & G(b) \end{array}$$

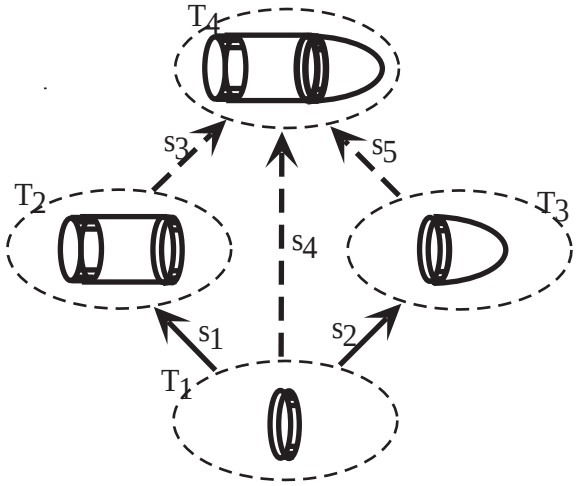

Figure 2: **A colimit builds complex objects from simple ones via commutative diagrams.**

That is, the morphisms $G(f) \circ \gamma_a : F(a) \longrightarrow G(b)$ and $\gamma_b \circ F(f) : F(a) \longrightarrow G(b)$ are actually one and the same, $G(f) \circ \gamma_a = \gamma_b \circ F(f)$. In simple terms, a natural transformation between two functors provides a way to switch from one mapping of a structure to another in a way that is interchangeable with the two images of any morphism. This concept is the glue that holds functorial implementations together.

3 Categories of concepts and neural components

The category **Concpt** has concepts as objects and morphisms $s: T_i \longrightarrow T_j$ that specify the manner in which T_i is a subconcept, or logical part, of T_j. We can view concepts mathematically as theories in formal logic, with sorts, operations and axioms as mentioned earlier. Besides mapping sorts and operations, a theory morphism has the following property: If A is an axiom of T_i, its image $s(A)$ is either an axiom of T_j or is provable from the axioms of T_j. The composition of morphisms in **Concpt** is a chaining-together of subconcept relationships. If T_i is a subconcept of T_j and T_j is a subconcept of T_k, then T_i is a subconcept of T_k by virtue of the realization of the two relationships involving T_i, T_j and T_k, taken in proper sequence. Every object has an identity morphism $\mathtt{id}_{T_i}: T_i \longrightarrow T_i$, consisting of a superposition of T_i onto itself.

Figure 3 shows a commutative diagram defining a colimit, which in this category is a formalization of the assembling of concepts and morphisms into a new, more complex concept. In this case, the complex concept is that of a streamlined cylindrical object formed from components consisting of a cylinder and blunt cone joined by a circular flange. The morphism $s_1: T_1 \longrightarrow T_2$ expresses the spatial placement of the flange in the cylinder-plus-flange composite T_2. The morphism $s_2: T_1 \longrightarrow T_3$ expresses the spatial placement of the flange at the rim of the blunt cone T_3. Together, these three objects and morphisms form a diagram that expresses the joining process: Only one flange is to be present, so the cylinder and cone must join at the flange. This is captured mathematically by the other three morphisms and the apical object T_4 at the top of the figure, a configuration called a cocone. The dashed arrows leading to the apical object T_4 are really theory morphisms describing the three separate embeddings of the components in the finished object. These combine with the lower diagram to form a commutative diagram: $s_5 \circ s_2 = s_4 = s_3 \circ s_1$. This is a mathematical way of stating that the two embeddings of the flange in the finished object—one via the cylinder and the other via the blunt cone—are really the same.

The other important property of colimits is called *initiality*. There is not room to address this in detail here, but the idea is that many other cocones for the lower diagram in Figure 3 apply also to larger diagrams that contain it. The apical object T_4 of the colimit is a concept with the minimal amount of information required to encapsulate the information in the original diagram. A colimit for a diagram can be thought of as a structure that "completes" the diagram to a minimal commutative diagram containing it.

To define a category **Neural** mathematically, we use some readily-available items from **Set**. There

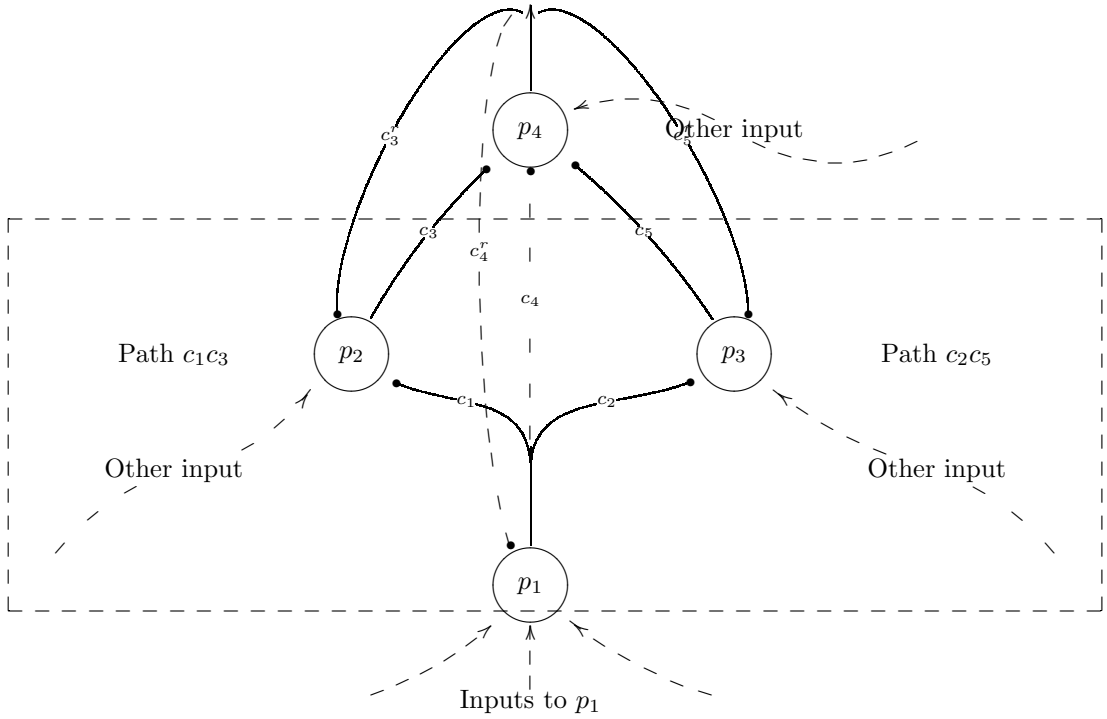

Figure 3: **Strong reciprocal connections enforce a commutative diagram.**

is not room here for all the details, but to summarize, a pre-architecture A has nodes $p_i^A\,(i \in P_A)$ and connections $c_k^A\,(k \in C_A)$, where P_A, C_A are sets of indices associated with A. The nodes have signal functions, thresholds θ^A_{i,n_i^A}, and conveniently-selected quantum level values for activation and outputs, $\theta^A_{i,0} < \theta^A_{i,1} < \theta^A_{i,2} < \ldots < \theta^A_{i,\ell} < \ldots < \theta^A_{i,\infty}$ and $\xi^A_{i,0} < \xi^A_{i,1} < \xi^A_{i,2} < \ldots < \xi^A_{i,\ell} < \ldots < \xi^A_{i,\infty}$, where $\xi^A_{i,\mu} = \phi_i^A(\theta^A_{i,\mu} - \theta^A_{i,n_i^A})$. Other values of θ_i, ξ_i occur, with $\xi_i = \phi_i^A(\theta_i - \theta^A_{i,n_i^A})$, but the information of most interest is which levels an output exceeds, $\xi_i > \xi^A_{i,\mu}$. This determines the instances of the concept represented by the object $(p_i^A, \xi^A_{i,\mu})$. An architecture is a pre-architecture together with state functions for network activity and weight adaptation; we call an instance of activity for an architecture an a-state and a weight configuration for it a w-state. Without loss of generality, we will focus on a single architecture and omit the superscript A; the morphisms of interest occur within a single architecture at a time. Also, we further simplify here by representing the objects as nodes, p_i, neglecting quantum levels. A morphism $m\colon p_i \longrightarrow p_j$ from object p_i to object p_j is the set of all priming states for a connection path $c_{k_1}c_{k_2}\cdots c_{k_n}$ with the respective objects as source and target nodes. A priming state for a path is an a-state in which the path weights and nodes are potentiated to transmit an output signal from p_i of sufficient strength to activate p_j. When we construct the functors of interest, with nodes as concepts, the priming of a path between two nodes represents an instance of a particular concept morphism, associated now with the path. Several paths can have the same priming states under the right conditions, and that facilitates commutative diagrams. The composition $(m' \circ m)\colon p_i \longrightarrow p_\ell$ of two morphisms $m\colon p_i \longrightarrow p_j$ and $m'\colon p_j \longrightarrow p_\ell$ is the set of priming states for any path obtained by concatenating any two of their respective paths.

A convenient architecture for commutative diagrams in **Neural** has a reciprocal connection c_k^r from p_j to p_i for every connection c_k from p_i and p_j such that both have the same polarity, excitatory or inhibitory. Initially, there are many reciprocal pairs of both polaritities incident upon a given node p_i. The connection weights w_k, w_k^r in an inhibitory reciprocal pair are fixed and large enough in magnitude that the set of nodes connected to p_i via inhibitory connections form a highly competitive subnetwork N_i. The N_i are organized via adaptive excitatory connections into clusters, some of which contain the input nodes associated with a sensor or internal pattern-generator. This arrangement supports the functorial mapping of

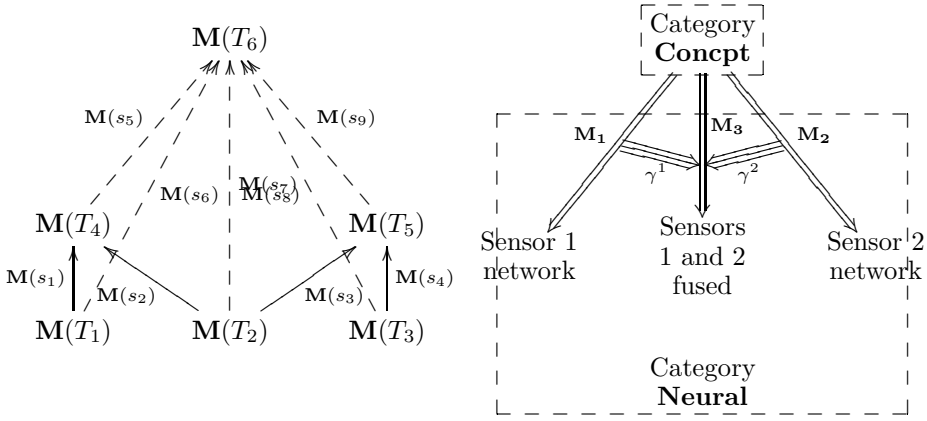

Figure 4: **(Left) A commutative diagram in** *Concpt* **maps to a commutative diagram in** *Neural* **via the functor** $M\colon Concpt \longrightarrow Neural$**. (Right) Functors connected by natural transformations implement a concept hierarchy in a multi-sensor system.**

the structure of **Concpt** into that of **Neural**. Initially, the excitatory weights are of small magnitude, with an arbitrary distribution. Based upon sensor input and network interactions, the current winner-take-all nodes among the N_i, being simultaneously active, undergo long-term potentiation to form strong reciprocal pairs of excitatory connections.

A commutative diagram for a colimit forms when a sufficently novel stimulus pattern appears at the input nodes. Several of the currently-active nodes can have strong excitatory connections from previous adaptation. These form a base diagram G, with the active nodes as objects and their active, hence, primed, excitatory interconnections participating in morphisms between the objects. A newly-recruited, active node p_j has its connections with the diagram objects strengthened. The resulting configuration of objects and morphisms forms a new diagram G_*. If the newly-strengthened weights w_k^r are large enough in magnitude to overcome any inhibitory inputs to the nodes in G, then G_* is commutative and therefore p_j and its input connections from the diagram G objects correspond to a cocone. The example of Figure 3 illustrates the priming connections that ensure the commutativity of a diagram formed in this manner, so that $(m_3 \circ m_1) = m_4 = (m_5 \circ m_2)$.

Defining a functor $\mathbf{M}\colon \mathbf{Concpt} \longrightarrow \mathbf{Neural}$ at any stage of learning involves an appropriate architecture organized with excitatory and inhibitory reciprocal connections as described above. Each **Concpt** morphism $s_\alpha\colon T_\mu \longrightarrow T_\nu$ maps to an appropiate **Neural** morphism $\mathbf{M}(s_\alpha)\colon \mathbf{M}(T_\mu) \longrightarrow \mathbf{M}(T_\nu)$ for some pair of network nodes p_i, p_j in the architecture such that $p_i = \mathbf{M}(T_\mu)$ and $p_j = \mathbf{M}(T_\nu)$. The functorial property $\mathbf{M}(s_k) \circ \mathbf{M}(s_{k'}) = \mathbf{M}(s_k \circ s_{k'})$ shows how to build an architecture incrementally. The need to implement commutative diagrams has in fact led to the architectural design scheme with reciprocal excitatory and inhibitory connections serving as the connections structure for competitive subnetworks and clusters discussed earlier. Nodes implementing several somewhat similar concepts, and responding to a single sensor, define the subnetworks N_i; single-connection morphisms occur between the objects in different N_i. As learning continues, more complex objects are formed, and more and more subnetworks become involved to represent competing complex objects. Each apical object $\mathbf{M}(T_\nu)$ of a cocone (illustrated by the very simple case in Figure 3) corresponds to one of the competing nodes in a subnetwork N_j. It has morphisms from the objects $\mathbf{M}(T_\mu)$ in other subnetworks N_i, which form a diagram along with morphisms supported by connection paths between these nodes in the N_i.

There is a different functor not only for each stage of learning, but for each sensor and combination of sensors. We can design an architecture to learn concepts relevant to any particular collection of sensors, or concepts which involve many sensors for their full expression but must be implemented with a limited sensor array for economy reasons. For example, suppose sensors S_1 and S_2 are available. Separate functors $\mathbf{M_1}$ and $\mathbf{M_2}$ can be used to represent the same state of learning in an architecture, but restricting concept implementations to a single sensor in each case. Thus, each concept and morphism are represented twice—once for sensor S_1 and once for sensor S_2. Each functor maps to a separate set of clusters of competitive subnetworks N_i; thus, two disjoint subnetworks of the architecture implement the same concepts for two

separate sensors. How are these to be unified, so that the two sensors can be exploited in forming these concepts? The answer lies in natural transformations $\gamma^{\mathbf{1}}\colon \mathbf{M_1} \longrightarrow \mathbf{M_3}$ and $\gamma^{\mathbf{2}}\colon \mathbf{M_2} \longrightarrow \mathbf{M_3}$ from $\mathbf{M_1}$ and $\mathbf{M_2}$ to a third functor $\mathbf{M_3}$, as shown in Figure 4. These are implemented by `Neural` morphisms via reciprocal connections as usual. Yet another distinct set of clusters of competetive subnetworks is used for the image of $\mathbf{M_3}$; however, $\mathbf{M_3}$ has no sensor. Instead, each triple of objects $\mathbf{M_i}(T_\mu)\,(i = 1,2,3)$ is connected by the two individual morphisms $\gamma^{\mathbf{1}}_{\mathbf{T}_\mu}\colon \mathbf{M_i}(T_\mu) \longrightarrow \mathbf{M_3}(T_\mu)\,(i = 1,2)$. This connects the three functorial images of the concept T_μ. The defining property of a natural transformation specifies that the connections between clusters that support the natural transformation morphisms are the appropriate ones to unify the two sensors in the dual-sensor representation of functor $\mathbf{M_3}$. This is true because each morphism $s_\alpha\colon T_\mu \longrightarrow T_\nu$ specifies a subconcept relationship, and the functorial images $\mathbf{M_i}(s_\alpha)\colon \mathbf{M_i}(T_\mu) \longrightarrow \mathbf{M_i}(T_\nu)\,(i = 1,2,3)$ preserve that relationship. For example, there are two morphisms available from the sensor- S_1-based representation $\mathbf{M_1}(T_\mu)$ of T_μ to the desired fused, dual-sensor representation $\mathbf{M_3}(T_\nu)$ of T_ν: the compositions $\gamma^1_{T_\nu} \circ \mathbf{M_1}(s_\alpha)$ and $\mathbf{M_3}(s_\alpha) \circ \gamma^1_{T_\mu}$. Each of these morphisms has its associated connection paths. For consistency, however, we want them to have the same priming states, that is, to be alternate paths associated with the same morphism. But this is just the defining requirement $\gamma^1_{T_\nu} \circ \mathbf{M_1}(s_\alpha) = \mathbf{M_3}(s_\alpha) \circ \gamma^1_{T_\mu}$ of the natural transformation γ^1. The same holds for γ^2, corresponding to sensor S_2.

The categorical model, with functors from a category of concepts to a category of neural network components and natural transformations between these functors, provides a mathematical model for neural structures consistent with concept-subconcept relationships. Colimits of diagrams show how concepts can be combined, and how a concept can be re-used many times in forming more complex concepts. Functors map commutative diagrams to commutative diagrams, capturing this aspect of the colimit structure. Natural transformations express the fusion of single-mode sensor representations of concepts in the same neural architecture, connecting the different implementations of the concept hierarchy at all levels and in a consistent fashion. This mathematical model appears to be compatible with a model of the primate brain proposed by Damasio[1]. We hope that others will find this model interesting and useful for neural network analysis and design.

References

[1] A. Damasio. Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition. *Cognition*, 33:25–62, 1989.

[2] J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.

[3] M. J. Healy. Colimits in memory: category theory and neural systems. In J. S. Boswell, editor, *Proceedings of IJCNN'99: 1999 International Joint Conference on Neural Networks*, pages Session 3.4–no. 75. IEEE Press, 1999.

[4] B C Pierce. *Basic Category Theory for Computer Scientists*. MIT Press, 1994.