Knowledge Representation and Possible Worlds for Neural Networks

Michael J. Healy‡§
Corresponding author
13544 23rd Place NE
Seattle WA 98125
E-mail: mjhealy@ece.unm.edu

‡Department of Electrical and Computer Engineering
University of New Mexico
Albuquerque, New Mexico 87131
E-mail: tpc@ece.unm.edu

Abstract—The semantics of neural networks can be analyzed mathematically as a distributed system of knowledge and as systems of possible worlds expressed in the knowledge. Learning in a neural network can be analyzed as an attempt to acquire a representation of knowledge. We express the knowledge system, systems of possible worlds, and neural architectures at different stages of learning as categories. Diagrammatic constructs express learning in terms of pre-existing knowledge representations. Functors express structure-preserving associations between the categories. This analysis provides a mathematical vehicle for understanding connectionist systems and yields design principles for advancing the state of the art.

I. INTRODUCTION

A sizeable body of research has had the objective of explaining the semantics of neural computation, particularly with respect to learning ([2], [5], [9], [7], [3], [6], [1], [21], [23], [17], [24], [25], [12], [16], [11]). Much of this research is an attempt to decode the adapted connection-weight values as “if-then” rules that a network is supposed to have learned from its input stream. Formal-logic-like languages are often used to express rules unambiguously and sometimes with full mathematical rigor ([2], [11], [23]). Intuitively, the ability of a computational system to manipulate data in a systematic way is a manifestation of the knowledge represented in the system’s structural design and operation. In the present paper, we describe a more comprehensive and wholly mathematical semantic model that fully explores this notion. Instead of a partial expression of the semantics of a neural network in terms of a single body of logical rule forms, we propose a hierarchical structure of interrelated knowledge modules, or concepts, expressed as formal logic theories. The concepts are symbolic descriptions of objects and events, observable or imagined, at any arbitrary level of generality or specificity. They are organized as a many-threaded hierarchy ordered from the abstract to the specific. In this context, the semantics of a neural network can be expressed as an evolving representation of a distributed system of concepts, many of them learned from data via weight adaptation. The concepts declare the meaning of the operations performed by many of them learned from data via weight adaptation. The evolving representation of a distributed system of concepts, ordered from the abstract to the specific. In this context, other structural mappings describe the possible worlds, or scenes, events, and situations, that the concepts describe. Taken together, the concept, possible world, and neural representations enable a comprehensive, mathematically rigorous analysis of the capabilities of neural networks in terms of their input environments and task objectives. An initial result of this new semantic model is a set of design principles [15] some of which have been applied to achieve improved performance in a neural network application [14].

II. CATEGORY THEORY: A BRIEF INTRODUCTION

Introductions to category theory at varying levels of detail are available in [8], [18], [19], and [15]. It is a theory of mathematical structure based upon the notion of an arrow, or morphism—a relationship between 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 \( g \circ f : a \rightarrow c \) 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 \( f : a \rightarrow b \), \( g : b \rightarrow c \) and \( h : c \rightarrow d \), 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 \( \text{id}_a : a \rightarrow a \) such that for any arrows \( f : a \rightarrow b \) and \( g : b \rightarrow a \), \( \text{id}_a \circ g = g \) 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, \( (g \circ f)(x) = g(f(x)) \).
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 \( f : i \rightarrow a \). A terminal object \( t \) has every object \( a \) of \( C \) as the domain of a unique morphism \( f : a \rightarrow t \).

Every category \( C \) has an opposite category \( C^{\text{op}} \) obtained by reversing the arrows and the order of composition of \( C \). That is, for each arrow \( f : a \rightarrow b \) of \( C \) there is an arrow \( f^{\text{op}} : b \rightarrow a \) of \( C^{\text{op}} \), and for any \( g : b \rightarrow c \) that is also an arrow of \( C \), the composition \( g \circ f : a \rightarrow c \) in \( C \) is matched by a composition in \( C^{\text{op}} \) with order reversed, \( f^{\text{op}} \circ g^{\text{op}} : c \rightarrow a \). This category is associated with the Principle of Duality, by which a theorem in category theory yields an additional theorem through reversal of the arrows, the order of composition and the terms “initial” and “final”, and the addition or deletion of the prefix “co”.

These key notions are fundamental in the definition of limits and colimits. Colimits express mathematically the learning of more complex, specialized concepts through the re-use of simpler concepts already represented in the connection-weight memory of a neural network ([13], [15]). Here, we define limits, which express the learning of more abstract concepts; reversing the arrows and substituting “initial” for “terminal” and “cocone” for “cone” suffices to define colimits, the dual notion. 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 \( f_1 : a_1 \rightarrow a_3, f_2 : a_2 \rightarrow a_3, f_3 : a_1 \rightarrow a_4, f_4 : a_2 \rightarrow a_4 \). The diagram \( \Delta \) extends \( \Delta \) to a commutative diagram by adding a cone \( K \). The latter consists of an apical object \( b \) and morphisms \( g_i : b \rightarrow a_i \) such that \( f_1 \circ g_1 = g_3 = f_2 \circ g_2 \) and \( f_3 \circ g_1 = f_4 = f_4 \circ g_2 \), provided additional objects and morphisms satisfying these equations exist in \( C \). Cones for \( \Delta \) are the objects of a category \( \text{cone}_\Delta \) (whose morphisms are described in [15]). 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. A discrete diagram has objects \( a_i \) but no morphisms, in which case its limit is called a product. The dual notion is a colimit for a discrete diagram, called a coproduct.

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 \( C \), \( F(g \circ f) = F(g) \circ F(f) \), where \( \circ_C \) and \( \circ_D \) denote the respective compositions in \( C \) and \( D \); (2) for each object \( a \) of \( C \), \( F(1_{da}) = 1_{F(a)} \). 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 \( D \). Finally, natural transformations are structure-preserving mappings between functors; although they are not discussed here, we mention them because of their fundamental role in the semantic model [15].

III. FROM ONTOLOGY TO NEURAL NETWORK

The category with which we begin a study of ontologies is one we call Concept. For convenience, we express it as a category of theories in a formal logic ([10], [20]), and, hence, its objects (our concepts) are theories and its morphisms are theory morphisms. For illustration, our theories will be expressed in a first-order multi-sorted predicate calculus. A theory morphism \( s : T \rightarrow T' \) is a mapping of the syntax of a theory \( T \) into the syntax of a theory \( T' \), such that the axioms of \( T \) are mapped to axioms or theorems of \( T' \). This category has colimits for all diagrams and limits for some. We shall show by example that a colimit can be used to derive a more complex, specialized concept from the theories and morphisms in its base diagram, along with morphisms (the cocone leg morphisms) by which it relates to them. Dually, a limit derives a more abstract concept along with morphisms that relate it to its base diagram objects.

The concept of a triangle can be derived as a colimit for a diagram involving concepts about points and lines (the alternative of defining triangles in terms of angles would complicate the example). We can start with a rather abstract concept, a theory that defines lines in terms of undefined quantities, or primitives, called points. The definition is expressed using a predicate \( \tau \) on two arguments, a point and a line, and is true just in case the point “lies on” the line (see [4] for a discussion of geometries based upon this definition).
Concept T1
sorts Points, Lines
const p1: Points
const p2: Points
const p3: Points
op on: Points*Lines --> Boolean
Axiom Two-points-define-a-line is
forall(x, y:Points)
((x not= y) implies
(exists l:Lines)
(on (x, l) and on (y, l) and
((forall m:lines) (on (x, m) and
on (y, m)) implies (m = l )))
end
The sort symbols Points and Lines stand for the types of entities, points and lines, described by the theory. A multi-sorted logic such as this is a convenient alternative to an unsorted logic, in which the sorts would be replaced by expressions point (x) and line (l) using predicates point and line, and every formula would be required to incorporate expressions like this.

Notice that in addition to the definition of a line in terms of points, T1 also contains constants representing three arbitrary points p1, p2 and p3. Three more specialized concepts T2, T3 and T4 share T1 except with different names for the point constants. The latter concepts specialize T1 by including a line constant and associating two of the point constants with it via the on predicate. The association is specified in an additional axiom (name omitted) which also contains constants representing three lines, which happen to be identical. We reformulate all statements of T1 by term replacement in accordance with the symbol mapping to form their image statements in T2. As a consequence, the axiom of T1 relating points to lines maps to itself as an axiom of T2. The point constants p1, p2 and p3 map to the point constants pa1, pa2 and paext. In T2, pa1 and pa2 are associated with the line 1a via the on predicate, and paext is intended as a point “external to” 1a. Omitting the symbol maps for those symbols that are identical to their images, as is customary, morphism s1 is as follows:

Morphism s1: p1 ↦ pa1
p2 ↦ pa2
p3 ↦ paext

Morphisms s2: T1 → T3 and s3: T1 → T4 are similar to s1 but with different point constant targets pb1, pb2, pbext and pc1, pc2, pcext:

Morphism s2: p1 ↦ pbext
p2 ↦ pb1
p3 ↦ pb2

Morphism s3: p1 ↦ pc2
p2 ↦ pcext
p3 ↦ pce

Concepts T3 and T4 are identical, except with the names pa1, pa2, paext, 1a replaced with pb1, pb2, pbext, 1b in T3 and pc1, pc2, pcext, 1c in T4. A morphism s1: T1 → T2 maps the sort symbols Points and Lines and the on predicate symbol to the corresponding symbols in T2, which happen to be identical. We reformulate all statements of T1 by term replacement in accordance with

In T3, it is the images pb1 of p2 and pb2 of p3 that are associated with the line constant, 1b, while the image pbext of p1 is the “external” point. The associations are similarly reordered in T4; the point-to-line associations in each concept can be seen by noticing which points are the targets of those of T1 under the appropriate morphism and applying term substitution. A colimit for the diagram Δ with
objects $T_1, T_2, T_3, T_4$ and morphisms $s_1, s_2, s_3$ (which always exists in the category Concept) has a cocone as shown in Figure 2, with apical object $T_5$ and leg morphisms $\ell_1:T_1 \rightarrow T_5$, $\ell_2:T_2 \rightarrow T_5$, $\ell_3:T_3 \rightarrow T_5$, and $\ell_4:T_4 \rightarrow T_5$. With $\Delta$ as the base diagram, the defining diagram of the colimit, $\Delta$, is commutative, with

$$\ell_1 = \ell_2 \circ s_1 = \ell_3 \circ s_2 = \ell_4 \circ s_3.$$ 

(1)

The resulting colimit apical object, $T_5$, is as follows:

**Concept T5**

- sorts Points, Lines
- const p1: Points
- const p2: Points
- const p3: Points
- const la: Lines
- const lb: Lines
- const lc: Lines
- op on: Points+Lines -> Boolean

**Axiom Two-points-define-a-line is**

forall(x, y:Points) ((x not= y) implies (exists l:Lines) (on (x, l) and on (y, l) and ((forall m:lines) (on (x, m) and on (y, m) implies (m = l))))

on (p1, la) and on (p2, la) and (p1 not= p2)

on (p2, lb) and on (p3, lb) and (p2 not= p3)

on (p3, lc) and on (p1, lc) and (p3 not= p1)

end

Because the defining diagram $\Delta$ of the colimit is commutative, the apical concept $T_5$ is a “blending” or “pasting together” of $T_2, T_3$ and $T_4$ along their common sub-concept $T_1$. That is, for the equality (1) to hold, separate symbols of $T_2, T_3$ and $T_4$ that are images of the same symbol of $T_1$ under the three diagram $\Delta$ morphisms $s_1, s_2$ and $s_3$ must merge into a single symbol in the colimit apical concept $T_5$. To make this clear, each symbol in $T_5$ that is a merging of symbols has been assigned the name of the $T_1$ symbol underlying the merging. Thus, symbols such as Points, Lines and on appear in $T_5$, and appear only once, since they are mapped to themselves by each of the morphisms $s_1, s_2$ and $s_3$. The point constants p1, p2, p3 also appear. However, in $T_5$, each one represents a merging of two point constants from $T_2, T_3$ and $T_4$ and as a consequence appears in the definition of two different lines. In two of these concepts, the image of each single point constant appears in the definition of a line, but as a different point on a different line in each of the two concepts. In the third concept, it appears as an “external” point, not on the line named in that concept. For example, p1 in $T_1$ is mapped to p1 in $T_2$ via s1, to pbext in $T_3$ via s2, and to pc2 in $T_4$ via s3. In $T_5$, therefore, it forms the point p1 at the intersections of lines 1a and 1c, and lies external to line 1b. Because of the initiality of the colimit cocone, any other cocone for $\Delta$ is the codomain of a unique cocone morphism whose domain is the cocone containing $T_5$. Therefore, $T_5$ adds no extraneous information to that in $\Delta$.

Because it is more complex, the colimit apical object is more specific, or more specialized, than any of the concepts in its base diagram. Because it expresses the “pasting together” of the base diagram concepts around their shared sub-concepts, it expresses the concept relationships (morphisms) as well as the concepts in its base diagram. Because of the Colimit Theorem, the calculation of concept colimits can be automated. For example, the apical object $T_5$ and leg morphisms $\ell_1, \ell_2, \ell_3$, and $\ell_4$ above can be derived automatically from the objects and morphisms of the base diagram, $\Delta$. Other examples are given in [26] for an application of engineering software synthesis via category theory. Where they are available, limits can be calculated also, yielding less complex or abstract theories and their attendant morphisms. This facilitates the generalization of learned concepts to new contexts by extracting useful invariants from them. The ability to calculate concept colimits and limits suggests the ability to “flesh out” an ontology in an incremental fashion. This process can begin with a collection of concepts and morphisms describing the most basic properties of observable quantities and also any desired assumptions about the environment and the operation of a system within it. More specialized concepts and morphisms can be calculated as colimits, and more abstract ones as limits, through the re-use of concepts and morphisms already defined.

IV. NEURAL CATEGORIES AND FUNCTORS

Each neural network architecture $A$, together with an array $w$ of connection weight values, has an associated category $\mathbf{N}_{A,w}$. This category is a mathematical representation of the structure comprising the network connections, its current weights, and its potential activation states initiated by inputs given the weight array $w$. The objective in defining $\mathbf{N}_{A,w}$ is to determine to what extent a functor $M: \text{Concept} \rightarrow \mathbf{N}_{A,w}$ can be defined for a given pair $(A, w)$ mapping concepts and their morphisms to the objects and morphisms of $\mathbf{N}_{A,w}$. The ability to perform this analysis brings mathematical rigor to the understanding of knowledge representations in neural networks.

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 $p_i$. The set $\eta$ can be regarded as an interval of real values if the analysis at hand considers $p_i$ to have a real-valued signal function. However, this is not a requirement of the semantic theory. For example, complex signal functions (and complex connection weights) can be used in an analysis and $\eta$ can be a region in the complex plane. In any case, 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 $\eta'$ along with a specified output set for each intermediate
There are four neural morphisms \( m_1, m_2, m_3, m_4 \) defined by the single-connection paths \( c_1, c_2, c_3, c_4 \) between the neural objects \((p_1, \eta), (p_2, \eta'), (p_3, \eta''), (p_4, \eta^{(3)})\). The compositions \( m_3 \circ m_1 \) and \( m_4 \circ m_2 \) defined by the two signal paths from \((p_1, \eta)\) to \((p_4, \eta^{(3)})\) along \( c_1, c_3 \) and \( c_2, c_4 \) through \((p_2, \eta')\) and \((p_3, \eta'')\) are morphisms between \((p_1, \eta)\) and \((p_4, \eta^{(3)})\). Another morphism \( m_7 \) between the latter two objects is defined by the set containing both \((p_1, \eta)\), \((p_3, \eta'')\), and \((p_4, \eta^{(3)})\), since \( m_3 \circ m_1 = m_8 = m_4 \circ m_2 \).

In general, for two paths which meet end-to-end, one concatenates the end-to-end paths to obtain a single path, and forms the intersection of the two instance sets. Concatenating the single-connection paths \( \gamma_1 \) consisting of \((p_1, \eta), (c_1, (p_2, \eta'))\) and defining a morphism \( m_1 \), and \( \gamma_3 \) consisting of \((p_2, \eta'), (c_3, (p_4, \eta^{(3)}))\) and defining a morphism \( m_3 \), yields a two-connection path \( \gamma_5 \) consisting of \((p_1, \eta), (c_1, (p_3, \eta''), (c_3, (p_4, \eta^{(3)}))\) and defining a morphism \( m_5 \). The instance set of \( \gamma_5 \) and, hence, of \( m_5 \), is \( U_{\gamma_5,w} = U_{\gamma_1,w} \cap U_{\gamma_3,w} \). We also define composition in this manner, so that \( m_5 = m_3 \circ m_1 \). Similarly, another two-connection path is evident from \((p_1, \eta)\) to \((p_4, \eta^{(3)})\) via \((p_3, \eta'')\) and defining a morphism \( m_6 = m_4 \circ m_2 \). If we can establish that \( U_{\gamma_6,w} = U_{\gamma_6,w} \), the diagram formed by \( m_3, m_2, m_3, m_4 \) is commutative, \( m_3 \circ m_1 = m_7 = m_4 \circ m_2 \), where \( \Gamma_7 \) is the two-path set \( \{ \gamma_7, \gamma_6 \} \), with \( U_{\gamma_7,w} = U_{\gamma_7,w} \cap U_{\gamma_6,w} \). The equality of path sets as shown is one of two alternatives in Figure 3. The other alternative, with \( m_8 \) in place of \( m_7 \), \( \Gamma_8 = \{ \gamma_5, \gamma_6 \} \), with \( U_{\gamma_8,w} = U_{\gamma_7,w} \cap U_{\gamma_6,w} \) has the path-sets not equal, \( U_{\gamma_7,w} \neq U_{\gamma_6,w} \), in which case the diagram is not commutative.

Notice that some weights of \( w \), and, hence, the network’s overall response to its inputs, may be changing under the influence of an instance \((\theta, e)\) of a morphism. The network activity and consequent weight-change resulting from \((\theta, e)\) result in a transition within \( A \) to a different stage of learning, where the weight array has transitioned from \( w \) to \( v \). We express this as a change from a category \( N_{A,w} \) to a different category \( N_{A,v} \). There can be many categories \( N_{A,v} \) associated with a single morphism \( m_{\Gamma,w} \) of \( N_{A,w} \) because the instances \((\theta, e)\) in \( U_{\Gamma,w} \) can be associated with many different transitions.

The mathematical machinery associated with functors of the form \( M: \text{Concept} \rightarrow N_{A,w} \) enables the study of the incremental changes in the network’s knowledge representation. The determination of the colimit and limit derivations possible in the neural categories associated with each stage of learning reveal how and at what stage these constructs are or can be represented in an existing or proposed neural architecture. This provides a mathematical semantic analysis of the connectionist learning performed by the network as it advances in both directions — specialization and abstraction — beginning at the sensor-percept level. If the derivation of a functor \( M: \text{Concept} \rightarrow N_{A,w} \) is blocked for a pair \((A, w)\), we have identified a problem in forming a knowledge representation at the stage of learning represented by \( w \). Since functors are many-to-one mappings on objects and morphisms, it is not necessary for an explicit representation of all concepts to be available — which is fortunate, since the category Concept is infinite and any stage of learning cannot be expected to have absorbed all knowledge in any case. Since many concepts and morphisms can be “compressed” in certain network regions by the many-to-one mapping, the inability to construct a functor signifies either a shortcoming in the design of \( A \) or that a transition to weight...
array \ w \ driven \ by \ an \ input \ stream \ is \ unrealistic: \ the \ array \\ w \ cannot \ be \ expected \ to \ occur, \ and \ if \ it \ has \ occurred, \ the \ problem \ is \ with \ A. \ If \ we \ can \ determine \ that \ a \ functor \ cannot \ exist \ for \ most \ or \ perhaps \ even \ for \ any \ w \ that \ might \ result \\ from \ adaptation \ in \ A, \ the \ architecture \ has \ a \ clear \ deficit \ in its \ ability \ to \ represent \ its \ environment. \ On \ the \ other \ hand, \\ we \ can \ apply \ the \ analysis \ to \ determine \ an \ A \ that \ is \ capable \\ of \ learning \ a \ knowledge \ representation \ to \ the \ extent \ desired.

There \ is \ yet \ more \ to \ be \ obtained \ from \ the \ use \ of \ category \\ theory. \ This \ derives \ from \ an \ analysis \ of \ the \ “possible \ worlds” that \ can \ occur \ in \ the \ environment \ of \ A. \ These \ worlds, \ represented \ by \ the \ theories \ of \ Concept, \ are \ the \ subject \ of \ the \ next \ section.

V. MODEL SPACES

Each concept \ T \ has \ an \ associated \ space \ of \ models, \\ \( \mathcal{Mod}(T) \). \ A \ model \ is \ a \ mathematically-expressed \ situation \ or \ system \ of \ entities, \ a \ “possible \ world” \ described \ by \ T, \ that \ is, \ a \ system \ that \ satisfies \ the \ axioms \ of \ T. \ For \ example, \\ a \ model \ of \ the \ theory \ \( T_3 \) \ in \ the \ colimit \ example \ consists \ of \ an \ arrangement \ of \ entities \ corresponding \ to \ the \ sorts \ and \ operations \ and \ satisfying \ the \ theory \ axioms, \ with \ the \ \( p_1, p_2, p_3, 1a, 1b, 1c \) \ serving as labels \ for \ three \ distinct \ entities \ considered \ to \ be \ points \ and \ three \ distinct \ entities \ considered \ as \ lines. \ What \ are \ these \ entities? \ An \ obvious \ example \ is \ one \ constructed \ using \ points \ and \ straight \ lines \ in \ the \ Euclidean \ plane. \ Another \ can \ be \ constructed \ on \ the \ surface \ of \ a \ sphere, \ where \ lines \ are \ great \ circles. \ Yet \ another \ can \ be \ constructed \ from \ the \ grid \ of \ pixels \ in \ a \ video \ image \ display, \ with \ pixels \ as \ points \ and \ with \ specific \ rules \ for \ identifying \ sets \ of \ pixels \ as \ lines \ (such \ as \ rows, \ columns, \ and \ diagonals \ at \ all \ orientations \ in \ the \ pixel \ grid).

Still \ another \ is \ any \ set \ \( S \) \ of \ people \ such \ that \ any \ two \ members \ of \ \( S \) \ share \ a \ unique \ set \ \( L \) \ of \ attributes; \ if \ the \ people \ are \ the \ points, \ those \ sharing \ the \ attributes \ in \ \( L \) \ (e.g., \ dark \ hair, \ height \ between \ 5 \ feet \ 6 \ inches \ and \ 6 \ feet) \ form \ a \ line. \ Any \ other \ line \ \( L' \) \ must \ contain \ an \ attribute \ which \ is \ shared \ by \ at \ most \ one \ point \ (person) \ which \ (who) \ shares \ the \ attributes \ in \ \( L \).

The \ variety \ of \ such \ examples \ is \ infinite. \ However, \ a \ model-theoretic \ formulation \ demands \ that \ the \ entities \ in \ any \ of \ these \ examples \ belong \ to \ a \ mathematical \ system \ of \ some \ kind. \ This \ provides \ a \ means \ of \ studying \ the \ structure \ imposed \ upon \ a “world” \ by \ a \ theory. \ Conversely, \ given \ certain \ models, \ theories \ can \ be \ derived \ to \ describe \ them.

In traditional model theory \ the \ systems \ are \ sets \ with \ structure. \ For \ a \ multi-sorted \ logic \ such \ as \ that \ used \ in \ Section \ III, \ the \ mathematical \ structure \ of \ a \ set \ model \ \( \sigma \) in \ \( \mathcal{Mod}(T) \) \ incorporates \ a \ multiplicity \ of \ sets \ and \ functions, \ the \ objects \ and \ morphisms \ of \ the \ category \ \( \text{Set} \). \ For \ each \ sort \ \( u \) \ of \ \( T \), \ there \ is \ a \ set \ \( u_\sigma \). \ For \ each \ operation \ \( p \) \ of \ \( T \), \ where \ \( p \) \ is \ given \ the \ form \ \( p: u u \rightarrow u' \), \ there \ is \ a \ function \ \( p_\sigma: u_\sigma \rightarrow u'_\sigma \) \ mapping \ every \ member \ of \ the \ set \ \( u_\sigma \) \ interpreting \ the \ sort \ \( u \) \ to \ some \ member \ of \ the \ set \ \( u'_\sigma \) \ interpreting \ the \ sort \ \( u' \). \ A \ predicate \ is \ expressed \ as \ an \ operation \ \( p \) \ such \ that \ the \ sort \ \( u' \) \ is \ \( \text{Boolean} \), \ so \ that \ \( u'_\sigma \) \ is \ the \ set \ of \ \( \text{Boolean} \ \) \ values \ \( T, F \). \ For \ each \ constant \ \( c \) \ with \ sort \ \( u \), \ there \ is \ a \\ specific \ member \ \( c_\sigma \) \ of \ \( u_\sigma \). \ Finally, \ each \ axiom \ of \ \( T \) \ must \ be \ valid \ for \ all \ quantities \ \( u_\sigma, p_\sigma, c_\sigma \). \ For \ example, \ a \ model \ \( \sigma \) \ in \ the \ Euclidean \ plane \ for \ any \ of \ the \ theories \ of \ the \ triangle \ example \ has \ \( \text{Points}_\sigma \) \ as \ the \ set \ of \ points \ of \ the \ planes \ and \ \( \text{Lines}_\sigma \) \ as \ the \ set \ of \ straight \ lines \ in \ the \ plane.

The \ operation \ on: \ \( \text{Points} \times \text{Lines} \rightarrow \text{Boolean} \ \) \ signifies \ a \ relation \ predicate—which \ would \ be \ interpreted \ by \ a \ subset \ of \ the \ cartesian \ product \ \( \text{Points}_\sigma \times \text{Lines}_\sigma \) \ in \ an \ unsorted \ logic \ model—is \ instead \ interpreted \ by \ a \ function \ \( p_\sigma: (\text{Points} \times \text{Lines})_\sigma \rightarrow \text{Boolean} \_{\sigma}, \) \ where \ \( (\text{Points} \times \text{Lines})_\sigma = \text{Points}_\sigma \times \text{Lines}_\sigma \). \ The \ axiom

\begin{align*}
\text{Axiom} \ Two-points-define-a-line \ is \\
\text{forall}(x, y: \text{Points}) \ \ (x \not= y) \ \ implies \\
\text{(exists l:Lines)} \ \ ((\text{on} \ (x, 1) \ \text{and} \ \text{on} \ (y, 1)) \ \text{and} \\
((\text{forall m:lines}) \ \text{on} \ (x, m) \ \text{and} \ \text{on} \ (y, m)) \ \text{implies} \ (m = 1))
\end{align*}

holds \ for \ all \ quantities \ \( x, y \in \text{Points}_\sigma \) \ satisfying \ the \ precondition \ \( x \not= y \): That \ is, \ there \ exists \ a \ unique \ member \ \( \text{Lines}_\sigma \) \ defined \ by \ \( x \) \ and \ \( y \). \ If \ lines \ are \ interpreted \ as \ sets \ of \ points \ that \ can \ be \ identified \ uniquely \ by \ any \ two \ of \ their \ members, \ then \ \( \text{Lines}_\sigma \) \ is \ a \ set \ of \ sets \ that \ satisfy \ this \ very \ demanding \ condition.

Whether \ it \ is \ defined \ to \ lie \ within \ the \ Euclidean \ plane, \ a \ pixel \ grid, \ or \ some \ other \ context, \ a \ model \ of \ a \ theory \ can \ be \ expressed \ as \ a \ collection \ of \ sets \ and \ functions. \ The \ many \ different \ models \ a \ theory \ \( T \) \ can \ have \ are \ objects \ in \ its \ model \ space \ \( \mathcal{Mod}(T) \). \ These \ ideas \ can \ be \ used \ to \ analyze \ mathematically \ the \ input \ environment \ of \ a \ neural \ network. \ Entities \ of \ interest \ are \ collected \ into \ sets \ which \ represent \ sorts. \ Relationships \ between \ the \ entities \ become \ functions, \ and \ entities \ of \ special \ interest \ become \ named \ elements \ of \ the \ sets, \ representing \ sorted \ constants. \ Since \ there \ can \ be \ many \ theories \ represented \ explicitly \ by \ objects \ \( (p, \eta) \) \ through \ a \ functor \ \( M: \text{Concept} \rightarrow \text{N}_A \), \ and \ each \ theory \ can \ have \ many \ different \ models \ in \ \( \text{Set} \), \ which \ we \ hypothesize \ have \ correspondents \ in \ instance \ sets \ \( U_{(p,\eta)} \) \ (and \ similarly \ for \ morphisms), \ a \ given \ entity \ or \ collection \ of \ entities \ in \ the \ environment \ can \ be \ represented \ in \ many \ different \ ways \ (as \ can \ relationships \ between \ the \ entities). \ The \ importance \ of \ our \ semantic \ theory \ is \ evident \ here, \ however, \ because \ the \ different \ representations \ of \ entities \ and \ their \ relationships \ are \ related \ as \ the \ theories \ are \ related \ via \ theory \ morphisms. \ In \ addition, \ the \ model \ spaces \ have \ structure \ relating \ the \ models. \ The \ next \ section \ will \ address \ the \ first \ point, \ by \ showing \ how \ theory \ morphisms \ have \ corresponding \ morphisms \ that \ map \ model \ spaces \ to \ model \ spaces.

VI. MODEL-SPACE MORPHISMS

Notice \ that \ a \ model \ \( \sigma \) \ in \ \( \mathcal{Mod}(T_1) \), \ while it \ must \ have \ specific \ members \ of \ \( \text{Points}_\sigma \) \ identified \ as \ interpretations \ for \ its \ point \ constants, \ is \ not \ required \ to \ have \ specifically \ identified \ members \ of \ \( \text{Lines}_\sigma \) \ because \ \( T_1 \) \ does \ not \ specify \ any \ line \ constants. \ On \ the \ other \ hand, \ specific \ lines \ serving \ in \ the \ roles \ of \ its \ three \ line \ constants \ must \ exist \ in \ any \ model \ \( \sigma' \) \ in
The foregoing implies the existence of a function \( \text{Mod}(\ell_1) : \text{Mod}(T_b) \rightarrow \text{Mod}(T_1) \) that preserves the model structure imposed by the constraints of \( T_1 \), mapped into \( T_b \) by \( \ell_1 \). In general, every theory morphism \( s : T \rightarrow T' \) has an accompanying model-space morphism \( \text{Mod}(s) : \text{Mod}(T') \rightarrow \text{Mod}(T) \), where for every model \( \sigma' \) in \( \text{Mod}(T') \) there exists a unique model \( \sigma = \text{Mod}(s)(\sigma') \) in \( \text{Mod}(T) \). This relationship between concept and model-space morphisms is illustrated in Figure 4. A further insight can be found in the analysis of model spaces and their morphisms \( \text{Mod}(s) : \text{Mod}(T') \rightarrow \text{Mod}(T) \), which are in correspondence with concepts and their morphisms \( s : T \rightarrow T' \). We postulate that there is a correspondence between the instances of an object \( M(T) = (p_1, \eta) \) of \( N_{A,w} \) and the models in \( \text{Mod}(T) \). Similarly, we postulate that there is a correspondence between the inputs associated with each morphism \( M(s) : M(T) \rightarrow M(T') \) and the mapping of models through \( \text{Mod}(s) : \text{Mod}(T') \rightarrow \text{Mod}(T) \). This has great significance for neural networks, for it says that an instance of activity associated with an object \( (p_1, \eta) = M(T) \) must also be associated with an object \( (p_i, \eta) = M(T) \) when there is a Concept morphism \( s : T \rightarrow T' \), as illustrated in Figure 4. This yields a design principle for neural networks: There must be a mechanism to ensure that every instance of the codomain object of a neural morphism is also an instance of its domain object. The surest way to ensure this is for the associated neural network to have reciprocal connections, allowing activity to propagate backward from the node associated with the codomain object.

One aspect of this principle is that in a neural network \( (A, w) \) whose associated category \( N_{A,w} \) has colimits and limits, we can predict the network behavior associated with these diagrammatic structures as follows. First, an instance \( (\theta, e) \) of a colimit object must also be an instance of the objects in its base diagram. That is, a stimulus promoting feedforward signaling from a few of the diagram object nodes, or from another source, may cause a colimit object node to become active within the appropriate interval. This may or may not occur; the feedforward connection weights might be insufficient, the colimit object node may be in competition with other colimit object nodes as a representative of the same input \( e \), and so forth. However, if it does occur and the colimit object node’s activity persists (one of our neural transitions occurs), then the resulting feedback to the diagram must be sufficient to ensure that \( (\theta, e) \) is also an instance of all of the base diagram objects. For limits, the situation is reversed: The feedback occurs from each base diagram object node to the limit object node, forcing \( (\theta, e) \) to be an instance of the limit object when it is an instance of any one of its diagram objects; on the other hand, an instance of a limit object need not be an instance of any of its diagram objects; the limit object node may, however, provide enough input to make its base diagram object nodes serious competitors as possible representatives of the current input \( e \).

VII. THE STRUCTURE OF MODEL SPACES

Similarities between models of a theory give each model space an internal structure: A model \( \sigma \) of a theory \( T \) can sometimes be mapped in a structure-preserving fashion to another model \( \tau \) of \( T \), and these structural maps have a composition. A model homomorphism \( h : \sigma \rightarrow \tau \) is defined as follows. For each sort \( u \) of \( T \), there is a function \( h_u : u_\sigma \rightarrow u_\tau \) such that for each constant \( c : u \), the member \( c_\sigma \) of \( u_\sigma \) maps as \( h_u(c_\sigma) = c_\tau \subset u_\tau \); and for each operation \( p : u \rightarrow u' \) with its associated functions \( p_\sigma : u_\sigma \rightarrow u'_\sigma \) and \( p_\tau : u_\tau \rightarrow u'_\tau \) and each member \( a \) of \( u_\sigma \), we have \( h_u(p_\sigma(a)) = p_\tau(h_u(a)) \subset u'_\tau \). The composition of two homomorphisms \( h \) and \( h' \) is defined by the compositions of the individual functions, \( (h' \circ h)_u = (h'_u \circ h_u) \), where function composition is as usual given by \( (h'_u \circ h_u)(x) = h'_u(h_u(x)) \). The identity homomorphism \( \text{id}_\sigma : \sigma \rightarrow \sigma \) is easily shown to exist for each model \( \sigma \), where for arbitrary homomorphisms \( h : \mu \rightarrow \sigma \) and
k: σ \rightarrow \tau$, and any $a \in u_\mu$ and $b \in u_\sigma$, we have $((1_{a_\mu})_u \circ h_u)(a) = h_u(a) \in u_\sigma$ and $(k_u \circ (1_{a_\sigma})_u)(b) = k_u(b) \in u_\tau$. It also follows from the corresponding law for function composition that homomorphism composition is associative. Therefore, with models as its objects and homomorphisms as its morphisms, the model space $\text{Mod}(T)$ has been shown to be a category.

We have seen that model spaces are categories in their own right. A similar line of reasoning yields the fact that the model-space morphisms are functors between the model categories. A more comprehensive introduction, involving categorical logic and model theory, reveals yet more underlying structure (see any of [8], [20], [10]). If we can establish an explicit mathematical relationship between model categories and neural network input environments, we can achieve greater depth in analyzing the semantics of neural architectures and study the details of neural computation. This in turn will aid in enlarging the set of design principles for neural networks derived from the semantic theory. For example, an analysis of model categories and functors is expected to shed light on the relationship between the organization of the input environment of a neural network and the effect of sequences of inputs on learning via successive weight changes.

VIII. Conclusion

The mathematical semantic theory presented here can aid researchers in neural networks and neuroscience in achieving a deep understanding of neural computation. This understanding is made possible by viewing neural computation as driven by the derivation of a distributed ontology representing knowledge associated with the network’s environment. The current understanding achieved with the semantic theory has already provided design principles as well as insights for investigations in neural and cognitive science. Some insights derive from associating concept structures such as colimits and limits with neural architectures via the category Concept and functors of the form $M: \text{Concept} \rightarrow \mathbb{N}_A$: For example, colimit and limit derivations suggest a study of learning as the long-term potentiation of weights within connection structures that represent the defining diagrams. Another insight arises from the view of multi-regional or multi-sensor neural networks as knowledge-driven systems with a coherence requirement [15]. Model theory yields yet another insight, suggesting that an effective neural design must provide feedback so that concept morphisms can be correctly represented by including model-space morphisms within the connectionist structure. The colimit-limit and model-space morphism insights have been successfully applied [14]. Further depth in the theory, including an analysis of model spaces as categories, shows promise of yielding further insights.

REFERENCES


[19] S. Mac Lane. Categories for the Working Mathematician. Springer, 1971. This is the standard reference for mathematicians, written by one of the two co-discoverers of category theory. The other was S. Eilenberg.


