An Algebraic Generalization for Graph and Tensor-Based Neural Networks

CIBCB 2017 – Manchester, U.K.

Ethan C. Jackson\textsuperscript{1}, James A. Hughes\textsuperscript{1}, Mark Daley\textsuperscript{1}, and Michael Winter\textsuperscript{2}

August 24, 2017

The University of Western Ontario — London, Canada\textsuperscript{1}
Brock University — St. Catharines, Canada\textsuperscript{2}
Motivation
Neural Network Representations

No current standard language or mathematical framework for neural networks — no standard representation.
Neural Network Representations

No current standard language or mathematical framework for neural networks — no standard representation.

In computational neuroscience:

- CSA — Connection Set Algebra [5]
- NeuroML — XML-based model description [6]
- Custom representations
No current standard language or mathematical framework for neural networks — no standard representation.

In computational neuroscience:

- CSA — Connection Set Algebra [5]
- NeuroML — XML-based model description [6]
- Custom representations

In computer science:

- Tensor-based — TensorFlow[1], Theano[12], Keras [2]
- Graph-based — NEAT, HyperNEAT [7]
- Custom representations
Evolutionary Computation

NEAT

- Architecture and parameters evolved simultaneously
- Connections are encoded point-to-point

Tensor Frameworks

- Architectures typically hand-designed
- High level interfaces increasingly available
- Evolutionary algorithms beginning to be used [4]
Connectomics is concerned with studying brain function in relation to network organization and dynamics. See work by O. Sporns [10], [11].

- Self-similarity, fractal-like patterns
- At multiple levels of organization

The human genome does not encode each neural connection, point-to-point — it does not encode an adjacency list.
Find a common mathematical framework for existing tools that provides:

- A universal network description language
- Modular, generative operations
- Symbolic reasoning
- Model portability
- Self-similar or fractal-like representation
A feed-forward neural network architecture.
Example — Graph and Tensor

\[ T_1 = R_1 \cdot \iota_{F_1 \to F_1 \oplus F_2} + R_2 \cdot \kappa_{F_2 \to F_1 \oplus F_2} \]
\[ T_2 = \pi_{F_1 \oplus F_2 \to F_1} \cdot R_3 \cdot \iota_{A_1 \to A_1 \oplus B_1} + \rho_{F_1 \oplus F_2 \to F_2} \cdot R_5 \cdot \kappa_{B_1 \to A_1 \oplus B_1} \]
\[ T_3 = \pi_{A_1 \oplus B_1 \to A_1} \cdot R_4 \cdot \iota_{A_2 \to A_2 \oplus B_2} + \rho_{A_1 \oplus B_1 \to B_1} \cdot R_6 \cdot \kappa_{B_2 \to A_2 \oplus B_2} \]
\[ T_4 = \pi_{A_2 \oplus B_2 \to A_2} \cdot R_7 + \rho_{A_2 \oplus B_2 \to B_2} \cdot R_8 \]
\[ T_5 = R_9 \]

\[ \text{eval}(R_1 \ldots R_9) : \text{In} \to \text{Out} := T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdot T_5 \]

Example network as a HyperNEAT genome.

Example network as a Tensor expression.
The substitution of network connections by a repeated module.
A relational matrix describing a high-level neural network architecture.
Background
CSA is a mathematical framework for compactly describing neural architectures.

- Masks
- Value sets
- Elementary connection sets
- ‘Pseudo’-algebraic operations
- Functions for establishing connectivity
CSA is a mathematical framework for compactly describing neural architectures.

- Masks — Boolean relations
- Value sets — Real-valued vectors
- Elementary connection sets — Elementary relations
- ‘Pseudo’-algebraic operations — No formal properties
- Functions for establishing connectivity — Incomplete set
Classical or Boolean-valued relations

\[ R \subseteq A \times B \quad R : A \times B \rightarrow \{0, 1\} \]

Often interpreted as Boolean-valued matrices, e.g.:

\[
\begin{bmatrix}
A_1 & B_1 & 0 & 0 \\
A_2 & 1 & 1 & 1 \\
\vdots & \vdots & \ddots & \ddots \\
A_n & 1 & 0 & 1
\end{bmatrix}
\]
A relation algebra is an extended, residuated Boolean algebra.

Elementary relation algebraic operations:

- Union*
- Intersection*
- Composition*
- Complement*
- Difference*
- Converse

*Corresponds to set-theoretic operation.

Converse reverses order of all pairs.
A semiring is a very general algebraic structure. Its elements can be used to model connectivity between neurons.

\[ R = (\mathbb{R}, +, \cdot, 0, 1) \] with the standard addition and multiplication operations forms a semiring.

A matrix over the semiring \( R \):

\[
\begin{bmatrix}
1.2 & 8.2 & 0 \\
0.1 & 9.9 & 1.2 \\
& & \\
3.4 & 0.4 & 1
\end{bmatrix}
\]
A relation algebra is embedded in the matrix algebra over a large class of semirings that includes the field $\mathbb{R}$ and functions over $\mathbb{R}$.

- Requires minimal extension – a *flattening* operation
- Connectivity and values can be modelled using a single object
- A basic descriptive framework can be implemented as an extension of linear algebra
- Algebraic properties, other operations ‘for free’
Algebraic Framework and Extended Operations
The core algebraic framework consists of matrices over the field $\mathbb{R}$ within which relations are the matrices with $\{0, 1\}$-valued entries. We use special matrices called relational sums to compose modular network architectures.

Application of operation \textit{inject-top-left} to a matrix $R$. Left: a matrix $N \to N$. Right: a matrix denoted by $N \oplus A \to N \oplus B$. 
Extended Operations — Substitution

Using relational sums, we implemented extended operations to build modular networks.

Substitution of connections by a module. The resulting network could in turn be used as another substitution module.
Because relations are embedded in the matrix algebra, we can conveniently use matrices and relational reasoning to define operations.

\[
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0.2 & 0.3 & 0.6 \\
1 & 1 & 0 & 0 & 0 & 0.2 & 0.9 & 0.1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

- Substitution of one connection by a module
- Can prove that connectedness is preserved
Extended Operations — Substitution

Algebraic definition and reasoning about matrices using relational methods.

**Definition**

Let $N$ and $C$ be finite sets, $\text{Net} : N \rightarrow N$ and $S : C \rightarrow C$ be relations, and $(N_x, N_y) \in \text{Net}$. The substitution operation is defined by:

$$\text{subst}(\text{Net}, S, (N_x, N_y)) :=$$

$$\text{iTL}(\text{Net} - \{(N_x, N_y)\}, C, C) \cup$$

$$\text{iBR}(S, N, N) \cup \{(N_x, C_1), (C_n, N_y)\}$$

**Lemma**

Let $N$ and $C$ be finite sets, $\text{Net} : N \rightarrow N$ and $S : C \rightarrow C$ be relations, and $(N_x, N_y)$ be an element in $\text{Net}$. If $(C_1, C_n) \in S^+$ then $(N_x, N_y) \in \text{Net}S^+$, the transitive closure of $\text{Net}S$, where $\text{Net}S = \text{subst}(\text{Net}, S, (N_x, N_y))$. 
In our paper we show that:

- Neural network architecture can be described and manipulated using existing and custom relational operations.
- Operations are generic with respect to the set used to describe neural connectivity.
- Properties about operations can be proven using relation algebraic methods.
Next Steps
“Models will become more like programs ... will be grown automatically.” [3]
— François Chollet, author of Keras
Modular Neuroevolution

\[
\begin{align*}
R : N &\rightarrow N \\
N &= \text{In} \oplus F_1 \oplus F_2 \oplus A_1 \oplus A_2 \oplus B_1 \oplus B_2 \oplus C \oplus \text{Out}
\end{align*}
\]
A relational representation neural networks could have several advantages over other approaches.

- Multiple levels of organization
- Algebraic manipulation
- Conventional ANN modules
- Relational and functional modules
Modular Neuroevolution

- We are currently working on a neuroevolution framework based on Cartesian genetic programming (CGP) and existing work on CGP-based ANNs.
Modular Neuroevolution

- We are currently working on a neuroevolution framework based on Cartesian genetic programming (CGP) and existing work on CGP-based ANNs.
- The algebraic framework introduced in this work will be the basis for the genetic representation and operators.
We are currently working on a neuroevolution framework based on Cartesian genetic programming (CGP) and existing work on CGP-based ANNs.

The algebraic framework introduced in this work will be the basis for the genetic representation and operators.

Evolved networks will be a mix of de novo evolved modules and existing modules in the form of ANN layers, relational, and functional programs.
• We are currently working on a neuroevolution framework based on Cartesian genetic programming (CGP) and existing work on CGP-based ANNs.

• The algebraic framework introduced in this work will be the basis for the genetic representation and operators.

• Evolved networks will be a mix of *de novo* evolved modules and existing modules in the form of ANN layers, relational, and functional programs.

• The representation will be based on a mapping between algebraic expressions and a recursive, modular adjacency structure.
Thank you!


HyperNEAT: eplex.cs.ucf.edu/hyperNEATpage/


