Symbolic roles in vectorial computation
Vectorial encoding of symbolic structure

— in contrast to hybrid symbolic/vectorial representations

A single vector encodes (i) all the (vectorial) labels and (ii) the (discrete) structure in which they reside

Motivation: vector ~ neural state

Socher, Manning & Ng 2010
Vectorial encoding of symbolic structure

**TYPE:** Decompose structure into roles \{r_k\}

Approach 1: Absolute position
[Approach 2: Contextual (~ n-gram)]

Each \(r_k\) is assigned a vector encoding \(r_k \in R\) (linearly indep.)
— designed or learned

**INSTANCE:** Specific fillers for roles

Let \(f_k \in F\) (linearly indep.) be the label in role \(r_k\)
— \(f_k\) may be a vector encoding of a symbol \(f_k \in A\)
— designed or learned

**ENCODING:** \(v = \sum_k f_k \square r_k\)

Can be recursive:

\[\forall x \in \{0, 1\}^* \]

\[
\begin{array}{c}
\text{r}_x \\
\text{r}_{0x} \quad \text{r}_{1x}
\end{array}
\]

\[\text{r}_{0x} = \text{r}_0 \square \text{r}_x\]

\[R = \square_d R^{(d)}\]

Size: linear in number of roles

Tensor Product Representations (TPRs: 1990)
Summary: TPRs

TYPE: Decompose structure into roles $\{r_k\}$
Each $r_k$ is assigned a vector encoding $r_k \in R$ (linearly indep.)

INSTANCE: Specific fillers for roles
Let $f_k \in F$ (linearly indep.) be the label in role $r_k$
  $-$ $f_k$ may be a vector encoding of a symbol $\mathcal{f}_k \in A$

ENCODING: $v = \sum_k f_k \square r_k$

NOTE: Turns out to have important implications for grammatical theory
Computability theory over TPRs

What symbolic functions can be computed over TPRs using neural computation?

The functions in the following classes are computable in a linear neural network:

- \( = \) base of in-place symbol mappings
- \( C = \) closure under composition of [tree-manipulating primitives \( \cup B \)]
- \( P \sim \) “primitive recursive”

‘Primitive recursive’:
\[ C \subset P \]
\[ g, h \in P \Rightarrow f \in P \text{ when} \]
\[ f(s) = \begin{cases} g(s) & \text{if atom}(s) \\ h(f(\text{ex}_0(s)), f(\text{ex}_1(s))) & \text{otherwise} \end{cases} \]
Decoding TPRs

**INSTANCE** \( \mathbf{v} \): Inner product

\[
f_k = \mathbf{v} \cdot \mathbf{r}_k^+ \quad \text{— given } \{\mathbf{r}_k\}
\]

**SAMPLE** \( \{\mathbf{v}(\alpha)\} \): Generative model

Hypothesis: \( \{\mathbf{v}(\alpha)\} \) is a collection of TPRs, each encoding an instance of a symbol structure of a single type

\[
\mathbf{v}(\alpha) = \sum_k \mathbf{f}_k(\alpha) \square \mathbf{r}_k \quad \text{— where } \mathbf{f}_k(\alpha) \text{ encodes a symbol } \mathbf{f}_k(\alpha)
\]

Learning algorithms: derived from generative model

**TYPE:** What are \( \{\mathbf{r}_k\} \) and \( \{\mathbf{f}_k\} \)?

**INSTANCE:** For a given \( \alpha \),

which symbol \( \mathbf{f}_k(\alpha) \in A \) fills each role \( \mathbf{r}_k \) ?

**APPLICATION:** Decoding neuroimages of combinatorial stimuli (e.g., sentences, words).

Instance bindings \( \{\mathbf{f}_k(\alpha)/\mathbf{r}_k\} \) of stimuli are known, so only need learn the TYPE encoding.
Vectorial encoding of symbolic structure

**TYPE:** Decompose structure into roles \( \{r_k\} \)

**Approach 1:** Absolute position

**Approach 2:** Contextual (\( \sim n\)-gram)

Each \( r_k \) is assigned a vector encoding \( r_k \in R \) (linearly indep.)
— designed or learned

**INSTANCE:** Specific fillers for roles

Let \( f_k \in F \) (linearly indep.) be the label in role \( r_k \)
— \( f_k \) may be a vector encoding of a symbol \( f_k \in A \)
— designed or learned

**ENCODING:** \( v = \sum_k f_k \odot r_k \)

---

\[ R(\rightarrow, Y) \]

If \( X \) fills this role, vector encoding is: \( \mathbf{X} \sqsubseteq (R \sqsubseteq Y) \)
\( \mathbf{E} \mathbf{R} \sqsubseteq \mathbf{X} \sqsubseteq Y \)
(used in cognitive models)

**Approach 1:** [filler] \( \sqsubseteq \) [position]

**Approach 2:** [filler\(_1\)] \( \sqsubseteq \) [filler\(_2\)]