Microsoft Research

# Faculty Summit

**2014** 15TH ANNUAL

Microsoft Research Faculty Summit 2014 15TH ANNUAL

Paul Smolensky
Cognitive Science Dept
Johns Hopkins Univ

# Symbolic roles in vectorial computation

Microsoft

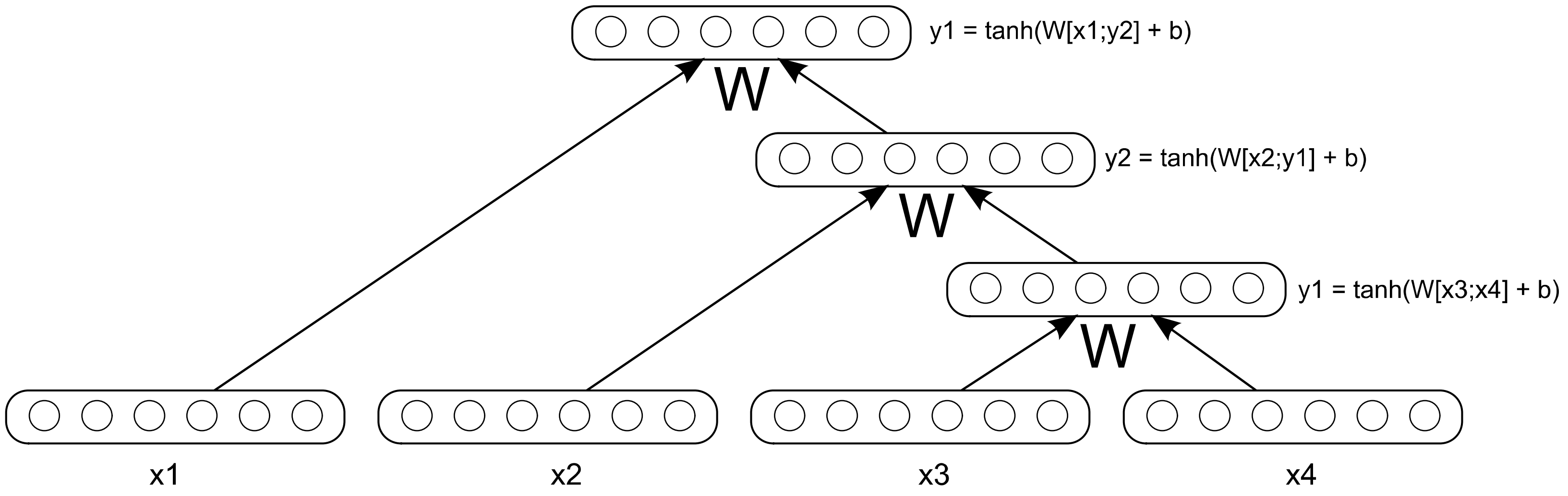
# Vectorial encoding of symbolic *structure*

## — 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 $\mathbf{r}_k \in R$ (linearly indep.)

— designed or learned

INSTANCE: Specific fillers for roles

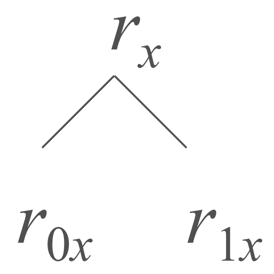
Let $\mathbf{f}_k \in F$ (linearly indep.) be the label in role $r_k$

— $\mathbf{f}_k$ may be a vector encoding of a symbol $\mathtt{f}_k \in A$

— designed or learned

ENCODING: $\mathbf{v} = \sum_k \mathbf{f}_k \otimes \mathbf{r}_k$

Can be recursive: $\forall x \in \{0, 1\}^*$




$$\mathbf{r}_{0x} = \mathbf{r}_0 \otimes \mathbf{r}_x$$

$$R = \oplus_d R^{(d)}$$

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 $\mathbf{r}_k \in R$ (linearly indep.)

## INSTANCE: Specific fillers for roles

Let $\mathbf{f}_k \in F$ (linearly indep.) be the label in role $r_k$

— $\mathbf{f}_k$ may be a vector encoding of a symbol $\mathtt{f}_k \in A$

ENCODING: $\mathbf{v} = \sum_k \mathbf{f}_k \otimes \mathbf{r}_k$

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:

- $\mathcal{B}$ = base of in-place symbol mappings
- C = closure under composition of [ tree-manipulating primitives ∪ $\mathcal{B}$ ]
- P ~ "primitive recursive"

'Primitive recursive':

C ⊂ P

$g, h \in \text{P} \Rightarrow f \in \text{P}$ when

$$f(s) = \begin{cases} g(s) & \text{if atom}(s) \\ h(f(\mathbf{ex}_0(s)), f(\mathbf{ex}_1(s))) & \text{otherwise} \end{cases}$$

# Decoding TPRs

## INSTANCE $\mathbf{v}$: Inner product

$\mathbf{f}_k = \mathbf{v} \cdot \mathbf{r}_k^{+}$ — 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

$\mathbf{v}^{(\alpha)} = \sum_k \mathbf{f}_k^{(\alpha)} \square\ \mathbf{r}_k$ — where $\mathbf{f}_k^{(\alpha)}$ encodes a symbol $\mathtt{f}_k^{(\alpha)}$

Learning algorithms: derived from generative model

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

INSTANCE: **For a given $\alpha$,**
**which symbol $\mathtt{f}_k^{(\alpha)} \in A$ fills each role $r_k$?**

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

Instance bindings $\{\mathtt{f}_k^{(\alpha)}/r_k\}$ 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 (~ *n*-gram)

Each $r_k$ is assigned a vector encoding $\mathbf{r}_k \in R$ (linearly indep.)

— designed or learned

INSTANCE: Specific fillers for roles

Let $\mathbf{f}_k \in F$ (linearly indep.) be the label in role $r_k$

— $\mathbf{f}_k$ may be a vector encoding of a symbol $\mathtt{f}_k \in A$

— designed or learned

ENCODING: $\mathbf{v} = \sum_k \mathbf{f}_k \,\square\, \mathbf{r}_k$

R(—, Y)

Role

If X fills this role, vector encoding is: $\mathbf{X} \,\square\, (\mathbf{R} \square \mathbf{Y})$

$\mathbf{E}\ \mathbf{R} \,\square\, \mathbf{X} \,\square\, \mathbf{Y}$

(used in cognitive models)

Approach 1: [filler] □ [position]

Approach 2: [$\text{filler}_1$] □ [$\text{filler}_2$]

Microsoft