Connectionist Symbol Processing

15-486/782: Artificial Neural Networks
David S. Touretzky
Fall 2006

What Is a Symbolic System?

- **Symbols** are arbitrary tokens.
- They combine to form **structures**.
- **Rules** govern derivation of new structures from old.

That's it!

Most of AI and CS is built on this framework.

Also, all of mathematics and logic.

Inference in Symbolic Systems

A **grammar** defines legal symbol structures.

A **production rule** in an expert system derives new working memory elements from old ones.

Logical deduction (Prolog, theorem provers).

Inductive reasoning (BACON).

Case-based reasoning.

Alpha-beta search.

All are part of the "symbolic paradigm."

What Do Neural Net Models Offer?

- Feature vector representations.
  - Dot product = similarity measure.
- A different approach to compositional structure.
- Different types of inference:
  - statistical learning (PCA, Boltzmann, Helmholtz, etc.)
  - nonlinear mapping (backprop)
  - parallel constraint satisfaction (Hopfield/Boltzmann)

A statistical, vector-based alternative to the symbolic framework.  More like the brain???
Feature Vector Representations

The representation of “chair” should be more similar to “table” than to “banana”.

Shouldn’t need axioms and rules to “deduce” this!

Ideally, the network should learn its own feature vector representations based on statistics.

See Hinton’s “family trees” model on next slide.

LSA (Latent Semantic Analysis):

Use SVD (like PCA) to derive “meaningful” feature vectors.

Hinton's Family Trees Network

Hinton's Family Trees Example

English:

Italian:

Hidden Layer Representation

15-498/782: Artificial Neural Networks  David S. Touretzky  Fall 2006
Hinton's Family Trees: Generalization Performance

- Domain:
  - 24 people (12 English, 12 Italian)
  - 12 relations (mother, father, husband, wife, son, daughter, aunt, uncle, brother, sister, nephew, niece)
  - 104 true relations over these family trees

- Training set:
  - Trained on 100 relations

- Test set:
  - Tested on 4 relations.
  - Got 3/4 right on one run, 4/4 right on another run.

Latent Semantic Analysis (LSA)

Method of constructing feature vectors for words.

- Build a frequency count table of words \times contexts.

- A context can be a sentence, a paragraph, or a document.

- Apply semantic/contextual weighting functions.

- Perform SVD (Singular Value Decomposition).

- Use resulting values as feature vectors for words or documents.

Inference by Nonlinear Mapping

- Elman's SRN (Simple Recurrent Network) predicts next word in a sentence.

- NETtalk text to speech system learns complex mapping.

- Pollack's recurrent network learned regular grammars.

- McClelland & Rumelhart's past tense model.

McClelland & Rumelhart's Past Tense Model

<table>
<thead>
<tr>
<th>Verb Root Encoding</th>
<th>weights (LMS rule)</th>
<th>Past Tense Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>flip --&gt; flipped</td>
<td>Regular forms</td>
<td></td>
</tr>
<tr>
<td>fib --&gt; fibbed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fit --&gt; fitted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>go --&gt; went</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sing --&gt; sang</td>
<td>Irregular forms</td>
<td></td>
</tr>
<tr>
<td>buy --&gt; bought</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bring --&gt; brought / brang / broughted</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Parallel Constraint Satisfaction

“The astronomer married the star.” (Pollack & Waltz)

More Parallel Constraint Satisfaction

• Hopfield nets for Traveling Salesman Problem
• Boltzmann machines
• Recurrent backprop nets that learn attractor states
• McClelland & Rumelhart word perception model.
Holographic Reduced Representations
(Tony Plate)

• Use large (N = 2000 elements) vectors.
  - Elements drawn independently from Normal(0, 1/N)
  - Vectors have roughly zero mean.
  - No similarity: $a \cdot b \approx 0$
  - Not a feature vector representation.

• Combine vectors using circular convolution operator.
  
  \[ C = a \times b \]

• Sort of like “cons” in Lisp: associates a with b.

Circular Convolution Properties

Commutative \( a \times b = b \times a \)

Associative \( a \times (b \times c) = (a \times b) \times c \)

Bilinear \( a \times (b \times c + y \times d) = b \times a \times c + y \times a \times d \)

Identity \( I \times a = a \)

Zero \( 0 \times a = 0 \)

Inverse \( a^{-1} \times a = I \)

Circular Convolution Operator

\[ z = x \times y \]

\[ z_i = \sum_{k=0}^{N-1} x_k y_{(i-k) \mod N} \]

Inverse of a Vector

A true inverse \( a^{-1} \) exists under certain conditions,
so that \( a^{-1} \times a = I \), but it may be numerically unstable.

If \( x \) is drawn from Normal(0,1/N) then it has an
approximate inverse \( x_i^T = x_{(i \mod N)} \).
Taking Apart a Pair

Given: \( a \times b \)
To get back \( b \), multiply by \( a^T \):

\[
a^T \times (a \times b) = (a^T \times a) \times b = I \times b = b
\]

Since \( a^T \) only approximates the inverse, we really get

\[
\hat{b} = b + \eta.
\]

Use a separate 'cleanup' process to eliminate noise:

\[
b + \eta \rightarrow b
\]

How to do cleanup?
- Associative memory
- Nearest-neighbor match

Convolution Preserves Similarity

\[
sim(a \times b, c \times d) \sim \sim(a, c) \cdot sim(b, d)
\]

Superpositions are similar to each of their components, to a decreasing degree as more patterns are added.

\[
sim(a, a+b) \geq \sim(a, a+b+c) \geq \sim(a, b)
\]

Similarity can be measured by dot product.

Encoding Propositions

Nouns: John, Mary, Spot
Verbs: bite, kiss
Roles: bite\textsubscript{agt}, bite\textsubscript{adj}, kiss\textsubscript{agt}, ...

'Spot bit Mary'

\[
C = \text{bite} + \text{bite}_\text{agt} \times \text{Spot} + \text{bite}_\text{obj} \times \text{Mary}
\]

Who was the biter? \( C \times \text{bite}_\text{agt} \sim \text{Spot} + \text{noise} \)
Who was bitten? \( C \times \text{bite}_\text{adj} \sim \text{Mary} + \text{noise} \)
What was the action? \( C \sim \text{bite} + \text{noise} \)
What did Spot do? \( C \times \text{spot}^T \sim \text{bite}_\text{agt} \)

Normalization

All vectors should be normalized to unit length:

\[
z = x \times y
\]

\[
\langle z \rangle = \frac{z}{\|z\|}
\]

Advantages:
- unbiased dot product similarity scores
- equal weighting of all component patterns
- reduced noise in decoded slot fillers

15-498/782: Artificial Neural Networks

David S. Touretzky
Fall 2006
Feature Representations

\[
\begin{align*}
John &= \langle \text{animal} + \text{human} + \text{id}_{John} \rangle \\
Jane &= \langle \text{animal} + \text{human} + \text{id}_{Jane} \rangle \\
Spot &= \langle \text{animal} + \text{dog} + \text{id}_{Spot} \rangle
\end{align*}
\]

\(John\) is more similar to \(Jane\) than to \(Spot\).

Recursive Composition

'Spot bit Jane, causing Jane to flee from Spot.'

\[
\begin{align*}
P_{\text{bit}} &= \langle \text{bite} + \langle \text{spot} + \text{jane} \rangle + \text{bite}_{\text{agt}} \times \text{spot} + \text{bite}_{\text{to}} \times \text{jane} \rangle \\
P_{\text{flee}} &= \langle \text{flee} + \langle \text{spot} + \text{jane} \rangle + \text{flee}_{\text{agt}} \times \text{jane} + \text{flee}_{\text{from}} \times \text{spot} \rangle \\
P &= \langle \text{cause} + \langle P_{\text{bit}} + P_{\text{flee}} \rangle + \text{cause}_{\text{ant}} \times P_{\text{bit}} + \text{cause}_{\text{caus}} \times P_{\text{flee}} \rangle
\end{align*}
\]

\(P\) contains a 'reduced description' of Jane being bitten.

Analogy Retrieval

Describe a situation in terms of predicates and objects.

'Spot bit Fred' = \(\text{bit}(\text{Spot, Fred})\)

Find an 'analogous' situation:

Same predicate? \(\text{bit}(\text{Fido, John})\)
Same objects? \(\text{saw}(\text{Fred, Spot})\)
Same predicate & objects? \(\text{bit}(\text{Fred, Spot})\)
Same object-roles? \(\text{bit}(\text{Spot, Jane})\)

Analogy Retrieval

- Ken Forbus' MAC/FAC is a symbolic analogy-making program.
- Use content vectors to summarize a description by counting the \# of occurrences of each object and predicate.
- Dot product of content vectors heuristically estimates similarity.
- High scoring descriptions are then explored symbolically to verify the analogy.
- But content vectors ignore structural similarity: "John bit Fido" treated the same as "Fido bit John".
A Better Retrieval Heuristic

Holographic Reduced Representations (HRRs) encode structural similarity:

\[
\langle \text{bite} + \langle \text{John} + \text{Fido} \rangle + \text{bite}_{\text{adj}} \times \text{John} + \text{bite}_{\text{adj}} \times \text{Fido} \rangle
\]

Structural Similarity

People are sensitive to structural similarity.

“Spot bit Jane, causing Jane to flee from Spot.”

LS (Literal Similarity)

Fido bit John, causing John to flee from Fido.

SF (Surface Features)

John fled from Fido, causing Fido to bite John.

CM (cross-mapped analogy) – roles switched

Fred bit Rover, causing Rover to flee from Fred.

Structural Similarity (cont.)

AN (analogy)

Mort bit Felix, causing Felix to flee from Mort.

\[
\text{a mouse} \rightarrow \text{a cat}
\]

FOR (first-order relations only)

Mort fled from Felix, causing Felix to bite Mort.

How Humans Handle Similarity

For LTM retrieval in humans:

\[
LS \Rightarrow CM \Rightarrow SF \Rightarrow AN \Rightarrow FOR
\]

### Episodes in long-term memory

<table>
<thead>
<tr>
<th>P</th>
<th>Spot bit Jane, causing Jane to flee from Spot.</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_{LS}</td>
<td>Fido bit John, causing John to flee from Fido.</td>
</tr>
<tr>
<td>E_{SF}</td>
<td>John fled from Fido, causing Fido to bite John</td>
</tr>
<tr>
<td>E_{CM}</td>
<td>Fred bit Rover, causing Rover to flee from Fred.</td>
</tr>
<tr>
<td>E_{AN}</td>
<td>Mort bit Felix, causing Felix to flee from Mort.</td>
</tr>
<tr>
<td>E_{FOR}</td>
<td>Mort fled from Felix, causing Felix to bite Mort.</td>
</tr>
</tbody>
</table>

#### Commonalities with probe

<table>
<thead>
<tr>
<th></th>
<th>Commonalities between episodes and probe structure</th>
<th>Similarity scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E_{\text{LS}}</td>
<td>E_{\text{SF}}</td>
</tr>
<tr>
<td>OK</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ERR</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HRR</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MAC</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

15-498/782: Artificial Neural Networks  David S. Touretzky  Fall 2006
Summary

- Connectionist symbol processing proposes a statistical, vector-based alternative to the "classic" symbolic paradigm.
- Issues:
  - feature vector encoding
  - compositionality (difficult)
  - nonlinear mapping
  - parallel constraint satisfaction
- Interesting ideas in all four areas.
- No one has figured out how to combine all four.
  Still in the early days of this research.