Constraint satisfaction

- Units represent hypotheses about parts of a problem
- Weights code constraints on how hypotheses can combine (i.e., the degree to which they are consistent or inconsistent)
- Possible solutions correspond to particular patterns of active units
- External input introduces bias to favor one possible solution over others

Temporal processing: Recurrent networks

Generalized Delta rule ("back-propagation")

Learning in networks with unrestricted connectivity:

**Back-propagation through time**

- Repeatedly update unit activations synchronously (first $n_j$, then $a_j$)
- Store entire activation history of each unit
- Attribute error to sending activations computed earlier in time

\[
\begin{align*}
n_j &= \sum_i a_i w_{ij} \\
a_j &= \frac{1}{1 + \exp(-n_j)}
\end{align*}
\]

Error $E = \frac{1}{2} \sum_j (t_j - a_j)^2$

Partial derivative of error with respect to weights:

\[
\frac{\partial E}{\partial w_{ij}} = \sum_j (t_j - a_j) a_j (1 - a_j)
\]

"Unfolding" a recurrent network into a feedforward one

Recurrent processing, pattern completion, and attractors

Recurrence can clean up noisy patterns into correct (clean) ones as long as input falls within correct basin of attraction
Temporal processing: Simple recurrent networks (Elman, 1990)

**Fully recurrent network**
- Computationally intensive to simulate
- Must update unit activities multiple times per input

**Simple recurrent network (SRN)**
- Adapt feedforward network to learn temporal tasks
- Computationally efficient but functionally limited compared to fully recurrent network

Sequential prediction tasks
- Input is a sequence of discrete elements (e.g., letters, words)
- Target is next item in the sequence
  - Self-supervised learning: environment provides both inputs and targets
- Network is guessing; cannot be completely correct but can perform better than chance if the input sequence is structured
- Given a particular sequence of past elements and current input, total error is minimized by generating the probabilities of next elements (i.e., their proportion of occurrence in this context across examples)
  - Across examples, each next element “votes for” its targets; result is average
- Example: Train on ABAC, ABCA and ACBB

<table>
<thead>
<tr>
<th>Time</th>
<th>Input</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A</td>
<td>0.0</td>
<td>0.67</td>
<td>0.33</td>
</tr>
<tr>
<td>1</td>
<td>B</td>
<td>0.5</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Letter prediction (Elman, 1990)**
- Network is trained to predict the next letter in text (without spaces)

Network discovers word boundaries as peaks in letter prediction error

**Word prediction (Elman, 1990)**
- Network is presented with sequence of words (localist representation for each)
- Sequence is constructed from 2-word or 3-word sentences
- No punctuation or sentence boundaries
- Trained to predict the next word within and across sentences
  - [analogous to predicting letters within/across words]

<table>
<thead>
<tr>
<th>Category</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOUN-HUM</td>
<td>man, woman</td>
</tr>
<tr>
<td>NOUN-ANIM</td>
<td>cat, mouse</td>
</tr>
<tr>
<td>NOUN-INANIM</td>
<td>book, rock</td>
</tr>
<tr>
<td>NOUN-AGRESS</td>
<td>dragon, monster</td>
</tr>
<tr>
<td>NOUN-FRAG</td>
<td>glass, plate</td>
</tr>
<tr>
<td>NOUN-FOOD</td>
<td>cookie, break</td>
</tr>
<tr>
<td>VERB-INTRAN</td>
<td>think, sleep</td>
</tr>
<tr>
<td>VERB-TRAN</td>
<td>see, chase</td>
</tr>
<tr>
<td>VERB-AGPAT</td>
<td>move, break</td>
</tr>
<tr>
<td>VERB-PERCEPT</td>
<td>smell, see</td>
</tr>
<tr>
<td>VERB-DESTROY</td>
<td>break, smash</td>
</tr>
<tr>
<td>VERB-EAT</td>
<td>eat</td>
</tr>
</tbody>
</table>
Learned word representations

Hierarchical clustering of hidden representations for words after training
- Network has learned parts of speech and (rudimentary) semantic similarity

Generalization to ZOG (man)

- Test network on novel input (ZOG) with no overlap with existing words, where ZOG occurs everywhere that MAN did
  - No additional training
- Network produces hidden representation for ZOG that is highly similar to that of MAN (based on context)