Connectionism

Phil/Psych 256

Chris Eliasmith
The Early Years

- Neuron doctrine (1900s)
  - Ramon y Cajal: many small units (neurons) connected in a vast and complex array

- Quantification:
  - Since the 1930s with mathematical descriptions of the functioning of this kind of system have been suggested

- Current attempts include:
  - Connectionism, Parallel Distributed Processing (PDP), Artificial Neural Networks (ANN), Computational/Theoretical Neuroscience
The Basics

- Simple units (‘neuron-like’):
  - fundamental unit of thinking is a simple computational unit, called a 'node' or 'neuron'.
  - artificial neurons compute a simple transfer function (e.g., linear, sigmoidal, binary, etc.)
  - a transfer function is a function that determines the neuron's output value given some input value

- Weighted connections:
  - to get interesting behaviors many nodes need to be connected together (hence the term 'connectionism')
  - these connections are weighted by some number
  - the weight determines the effect of the output of a previous neuron on the receiving neuron

- The combination of connections, weights, and transfer functions determines the behavior of the network
A Connected Network

Input

Weighted connection

Node/neuron

Output
The Fall

- Frank Rosenblatt (1960s) developed the Perceptron.
  - trained to learn interesting behaviors, like pattern recognition
  - Rosenblatt made strong claims regarding the potential of the machine

- Minsky and Papert (1969)
  - Book ‘Perceptrons’ showed serious limitations to this approach (the ‘XOR problem’).
XOR Problem

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Xor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The Rise

- Invention of ‘backpropagation’ (1980s)
- PDP Research Group (Rumelhart, McLelland, Smolensky, and others) solved the XOR problem
- Historical note: a paper by Sun-ichi Amari in the 70s in Japan had the solution but went unnoticed
- Backpropagation tells you how to adjust ‘hidden’ weights (next slide).
- Allows multi-layer networks (i.e. networks with more than one layer of computing nodes, unlike Perceptron)
- 3-layer neural networks can compute any Turing Machine function
- As a result, it is as computationally powerful as classical AI, although it excels at a different class of problems
Learning

- Quantitative result:
  - all (Turing) computable functions are accessible by changing only the weights

- Supervised Learning (most common):
  - an input (a dog picture) is presented to a network with random weights
  - the network produces some output ('sdfsa')
  - the right answer ('dog') is compared to that output ('sdfsa') and the errors are 'back-propagated' through the network, updating weights to minimize error
  - this is done many times, and eventually the network learns to associate the input dog with the output ‘dog’
Unsupervised Learning (aka self-organization):
- relies on statistics of the network's input to organize the output space into categories
- the network learns to pick out the 'most salient' patterns in the input

General features:
- Because the output is determined by the activity of many neurons, there are no explicit (higher-level) rules in these networks, only implicit ones.
- Networks usually learn to 'generalize' from the input they are presented. Thus, they can often classify novel input quite well
**Representations**

- **Localist representations:**
  - Each node in the network is assigned a determinate, symbolic meaning, e.g. 'Dog'.
  - The degree to which that node is active determines the degree to which that symbol is instantiated.
  - Localist networks are good for soft constraint satisfaction problems (text emphasis).
Representations (cont.)

- Distributed representations:
  - representations are 'scattered' across many nodes
  - the instantiation of a notion like 'dog' would be represented by the activity of many neurons (i.e., it's distributed across neurons) in the network
  - there can be different degrees of distribution in networks
    - e.g., nodes representing a dog could correspond to 'has brown hair', 'has nose', etc. (less distributed) or they could correspond to visual contrast at particular positions in retinal space (more distributed)
  - distributed networks are good for pattern recognition, and are more standard
Localist

Dog
Furry
Small
Cat
Legs

Distributed

Dog
Furry

Info Flow

Happy
Unconcerned
Sleek
Loner
Silly

Recurrent

OR

Info Flow

OR
Comparison

- Localist models
  - easier to interpret (why?)
  - useful for understanding higher, psych-level behaviors
  - But
    - very 'un-brain-like',
    - many of the same shortcomings as symbolic models (e.g., brittle, learning basic concepts is harder, modeling perceptual systems is unnatural, etc.).

- Distributed models
  - don't share these shortcomings (to the same degree)
  - But
    - quite difficult to interpret (what do weights/activities mean?)
    - success on modeling higher-level processes has been limited (although see Eliasmith and Thagard, 2001)
Past models

**NETTalk (Sejnowski and Rosenberg, 1986):**
- Backpropagation used to teach a distributed 2-layer feedforward network to pronounce words from any text using a voice synthesizer (97.5% accurate).
- The network’s state space is organized to correspond to the letter to phonemes mappings (‘a’ in 'fat' and 'a' in 'fate' are nearby but distinct).
- Compare DECTalk (formal rules, 78% accurate).

**TRACE (McLelland and Elman, 1986):**
- Completely pre-specified
- Used to recognize spoken words: not nearly as good as now, but much better than what was available.
- Accounted for interesting lexical effects (e.g. phoneme identification depends on lexical context - 'Christmas tapes').
Past models (cont.)

- **Rock/Mine detector (Gorman and Sejnowski, 1988):**
  
  A backpropagation network used to distinguish the sonar signals generated by rocks from those generated by mines (a task normally done by a human).

  The network had better performance than humans on the categorization task.

- **Past tense (Rumelhart and McLelland, 1986):**
  
  Perceptron network, mapped present onto past tense verbs.

  Showed the 'U' shaped learning that children show (i.e. first get irregulars correct, then generalize 'ed' even to irregulars (e.g. 'goed') and then get irregulars correct again).
More recent work

- Simple recurrent networks (SRNs; Ans, Rousset, French & Musca):
  - avoids catastrophic interference (forgetting what's been learned when new information arrives)
- Structured distributed representations (HRRs (Plate), spatter codes (Kanerva), tensor products (Smolensky), RAMMs (Pollack)) for supporting structure processing in distributed networks
- Unsupervised learning (particle filters, density estimation, etc.)
- Theoretical neuroscience…