Foundations of Artificial Intelligence
CS472/3
Lecture #31
Bart Selman

Slide CS472–1

• Neural networks cont.

Slide CS472–2
Topics
Type of network structure.
Type of representations.
Type of learning algorithms (and applicability).

Network Structure
Main distinction: feed-forward vs. recurrent.

Feed-forward: no cycles. Activation flows one direction —
from input layer via “hidden” layers to output layer.

Extreme (unlikely) example: input layer — retina cells /
output layer — muscle control cells.

Next figure: two (three?) layers. Two input units / two hidden
units one output unit
\[ a_5 = g(W_{3,5} \, a_3 + W_{4,5} \, a_4) \\
= g(W_{3,5} \, g(W_{1,3} \, a_1 + W_{2,3} \, a_2) + W_{4,5} \, g(W_{1,4} \, a_1 + W_{2,4} \, a_2))) \]
Activation passed from input to output. Does network have internal state? Corresponds to simple reactive agents. Much used! Good learning algorithms for classification / concepts.

Brain cannot be just a feedforward network!
Need (need short-term memory)

Brain has many feed-back connections.
brain is recurrent network.
Cycles!

---

**Hopfield Networks**

Much harder to analyze. Can capture internal state. (activation keeps going around) More complex agents.

Two main types:

- Hopfield networks.
- Boltzmann machines.
Hopfield Networks

symmetric connections \( W_{i,j} = W_{j,i} \)
output 0/1 only.
train weights to obtain associative memory
eg. store patterns (do figure).

It can be proven that an \( N \) unit Hopfield net can store up to \( 0.138N \) patterns reliably.
Note: no explicit storage. All in the weights.

Boltzmann machines

symmetric connections \( W_{i,j} = W_{j,i} \)
output 0/1 only but network in constant motion:
compute average output value of each node.
stochastic

has nice (but slow) learning algorithm. also closely connected to probabilistic reasoning belief networks.
details beyond the scope of this course.
Back to Feed-forward
input / output / hidden units.

perceptrons: no hidden units
multilayered

<table>
<thead>
<tr>
<th>Input Units</th>
<th>Output Units</th>
<th>Input Units</th>
<th>Output Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_j$</td>
<td>$W_{j,i}$</td>
<td>$O_i$</td>
<td>$I_j$</td>
</tr>
</tbody>
</table>

Perceptron Network | Single Perceptron

Slide CS472–11
Perceptrons

Remarkable learning algorithm: (Rosenblatt 1960)
if function can be represented by perceptron,
then learning algorithm is guaranteed to quickly converge
to the hidden function!

enormous popularity, early / mid 60’s

But analysis by Minsky and Papert (1969)
showed certain simple functions cannot be represented
(Boolean XOR)
Killed the field! (and possibly Rosenblatt (rumored)).

Linearly separable functions only

(a) Separating plane
(b) Weights and threshold

Slide CS472–13

Slide CS472–14
Mid eighties: comeback — multilayered networks
(Turing machine compatible)
learning procedures: **backpropagation**
Possibly one of the most popular / widely used learning methods today.
John Denker: “neural nets are the second best thing for learning anything!”

backprop and perceptron learning
Representations

How are concepts represented in the brain / neural net?

- local representations / grandmother cell
- distributed representations

Pros / Cons?
- distributed appeared to have won but
- in May 1997: UCLA researchers showed
  - single cell can learn a concept! (concept: facial expressions / a cell responding to “angry face”!)

Slide CS472–17