CS 730/730W/830: Intro AI

1 handouts: slides

asst 4 is due

730W blog entries were due

BackProp

Supervised Learning: Summary So Far

k-**NN** : distance function (any attributes), any labels

Neural network : numeric attributes, numeric or binary labels

Regression: incremental training with LMS

3-Layer ANN: non-linear wrt features

Inductive Logic Programming: logical concepts

The Three-layer Architecture

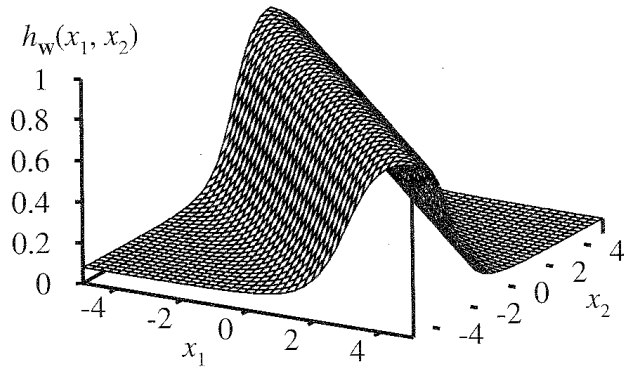
'hidden layer'

non-linear!

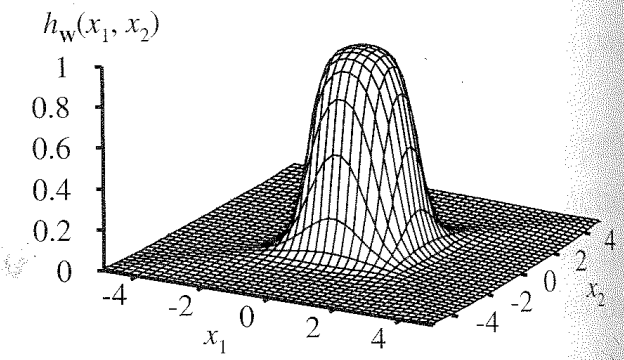
training: backwards error propagation

recurrence

Nonlinearity



(a)



(b)

Figure 18.23 (a) The result of combining two opposite-facing soft threshold functions to produce a ridge. (b) The result of combining two ridges to produce a bump.

Backwards Error Propagation

k inputs, j hidden units, i outputs

$g'(in_i)$ is derivative of activation function wrt input i

$$\begin{aligned} \Delta_i &= g'(in_i)(\hat{y} - y) \\ W_{j,i} &= W_{j,i} - \alpha a_j \Delta_i \end{aligned}$$

Backwards Error Propagation

k inputs, j hidden units, i outputs

$g'(in_i)$ is derivative of activation function wrt input i

$$\begin{aligned}
\Delta_i &= g'(in_i)(\hat{y} - y) \\
W_{j,i} &= W_{j,i} - \alpha a_j \Delta_i \\
\Delta_j &= g'(in_j) \sum_i W_{j,i} \Delta_i \\
W_{k,j} &= W_{k,j} - \alpha a_k \Delta_j
\end{aligned}$$

only locally optimal, dependence on structure

Break

- asst 4
- asst 5: data, tool, reference
- projects!

Decision Trees

Example: WillWait

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	$$$	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	$	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	$	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	$	*Yes*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	$$$	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	$$	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	$	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	$$	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	$	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	$$$	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	$	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	$	*No*	*No*	*Burger*	*30–60*	*Yes*

Figure 18.3 Examples for the restaurant domain.

Building a Decision Tree

DTLearn(examples, attributes, default)
if no examples, return default
if all same label, return it
$m \leftarrow$ majority label
if no attributes, return m
else
  $a \leftarrow$ choose attribute
  make node that branches on a
  remove a from attributes
  for each value v of a
    subtree $\leftarrow$ DTLearn(examples with $a = v$, attributes, m)
    add branch to subtree for v at node
  return node

Branching

want attribute that reduces uncertainity

Branching

want attribute that reduces uncertainity = entropy =

$$H(X) = -\sum_i P(x_i) \log_2 P(x_i)$$

where X is random var that takes value x_i with prob $P(x_i)$

Branching

want attribute that reduces uncertainity = entropy =

$$H(X) = -\sum_i P(x_i) \log_2 P(x_i)$$

where X is random var that takes value x_i with prob $P(x_i)$

information gain of atttribute A:

$$H(X) - \sum_{a \in A} P(a) H(X_a)$$

where X_a contains only examples with $A = a$

Branching

want attribute that reduces uncertainity = entropy =

$$H(X) = -\sum_i P(x_i) \log_2 P(x_i)$$

where X is random var that takes value x_i with prob $P(x_i)$

information gain of atttribute A:

$$H(X) - \sum_{a \in A} P(a) H(X_a)$$

where X_a contains only examples with $A = a$

stop when gain is small (χ^2 test, see p.705) or cross-validate

EOLQs

- What question didn't you get to ask today?
- What's still confusing?
- What would you like to hear more about?

Please write down your most pressing question about AI and put it in the box on your way out.
Thanks!