asst 10 posted

Supervised Learning

The Setting

labeled examples
hypothesis space
free parameters = degrees of freedom
classification vs regression
noise, overfitting

Using the k-Nearest Neighbors

majority. $k = 1$ gives Voroni cells

$$d(a, b) = \sqrt{\sum_i (a_i - b_i)^2}$$

normalize dimensions (divide by $\sqrt{\frac{1}{N}\sum_i (x_i - \bar{x})^2}$)
weight by distance?

$+$: robust to noise, choose k by easy cross-validation
$-$: time, memory, kd-tree, irrelevant features, sparse data in high d

Break

- asst 9
- asst 10
- wildcard vote!
- projects: posters Fri May 2 noon-2, papers Mon May 12 2pm

Regression

$$\hat{y} = \theta_0 f_0(x) + \theta_1 f_1(x) + \theta_2 f_2(x) + \dots$$

Regression

$$\hat{y} = \theta_0 f_0(x) + \theta_1 f_1(x) + \theta_2 f_2(x) + \dots$$

$$\hat{y} = \theta_0 + \theta_1 x$$

$$\hat{y} = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

$$\hat{y} = \theta_0 + \theta_1 \sin x$$

On-line Regression

$$\hat{y} = \theta f(x)$$

given sample x, y, want update to decrease $E = \frac{(\hat{y}-y)^2}{2}$:

On-line Regression

$$\hat{y} = \theta f(x)$$

given sample x, y, want update to decrease $E = \frac{(\hat{y}-y)^2}{2}$:

$$\theta_i \leftarrow \theta_i - \alpha \frac{\delta E}{\delta \theta_i}$$

On-line Regression

$$\hat{y} = \theta f(x)$$

given sample x, y, want update to decrease $E = \frac{(\hat{y}-y)^2}{2}$:

$$\begin{aligned}
\theta_i &\leftarrow \theta_i - \alpha \frac{\delta E}{\delta \theta_i} \\
\frac{\delta E}{\delta \theta_i} &= \frac{\delta}{\delta \theta_i} \frac{(\hat{y}-y)^2}{2} \\
&= (\hat{y}-y) \frac{\delta}{\delta \theta_i} (\hat{y}-y) \\
&= (\hat{y}-y) \frac{\delta}{\delta \theta_i} (\theta f(x) - y) \\
&= (\hat{y}-y) f(x)
\end{aligned}$$

On-line Regression

$$\hat{y} = \theta f(x)$$

given sample x, y, want update to decrease $E = \frac{(\hat{y}-y)^2}{2}$:

$$\begin{aligned}
\theta_i &\leftarrow \theta_i - \alpha \frac{\delta E}{\delta \theta_i} \\
\frac{\delta E}{\delta \theta_i} &= \frac{\delta}{\delta \theta_i} \frac{(\hat{y}-y)^2}{2} \\
&= (\hat{y}-y) \frac{\delta}{\delta \theta_i} (\hat{y}-y) \\
&= (\hat{y}-y) \frac{\delta}{\delta \theta_i} (\theta f(x) - y) \\
&= (\hat{y}-y) f(x) \\
\theta &\leftarrow \theta - \alpha (\hat{y}-y) f(x)
\end{aligned}$$

The LMS Procedure

$$\begin{aligned} \text{given } \hat{y} &= \theta x \\ \theta &\leftarrow \theta - \alpha(\hat{y} - y)x \end{aligned}$$

The LMS Procedure

$$\begin{aligned} \text{given } \hat{y} &= \theta x \\ \theta &\leftarrow \theta - \alpha(\hat{y} - y)x \end{aligned}$$

eg, for $x = \langle 1, x_1, x_2 \rangle$, the updates are:

$$\begin{aligned} \theta_0 &\leftarrow \theta_0 - \alpha(\hat{y} - y) \\ \theta_1 &\leftarrow \theta_1 - \alpha(\hat{y} - y)x_1 \\ \theta_2 &\leftarrow \theta_2 - \alpha(\hat{y} - y)x_2 \end{aligned}$$

$\alpha \approx 1/N$? or $100/(100 + N)$? or 0.1?
Δ rule, LMS weight update, Adaline rule, Widrow-Hoff rule, Perceptron rule
converges if data are linear
perceptron in finite time if linearly separable

The Three-layer Architecture

'hidden layer'
non-linear!
training: backwards error propagation
(recurrence)

Nonlinearity

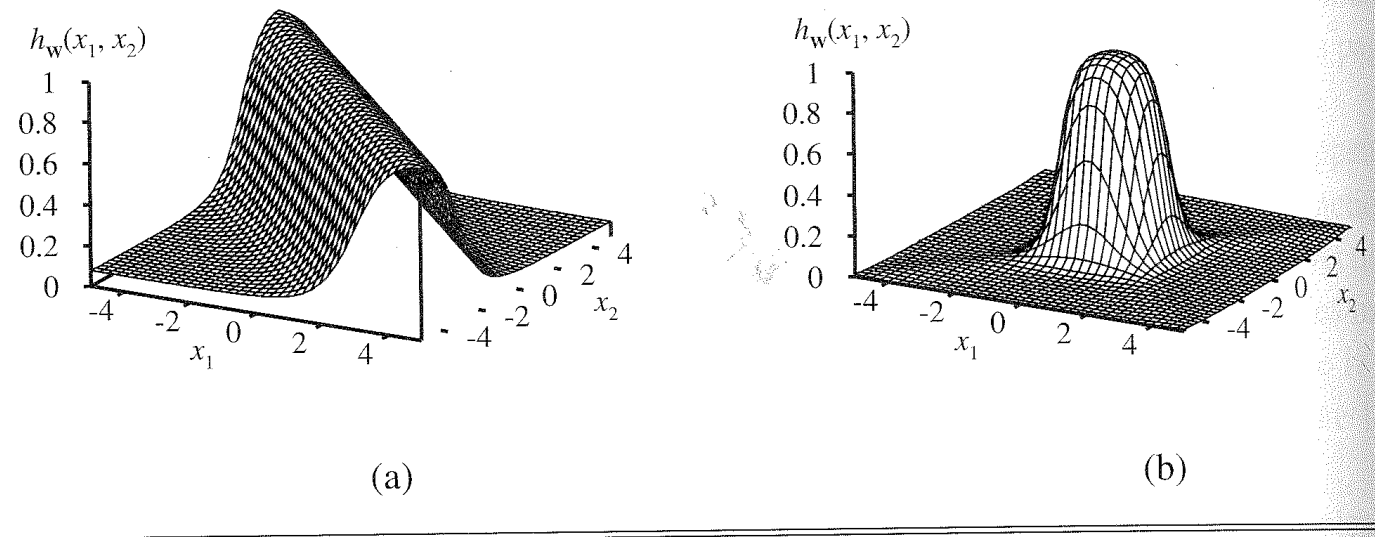

Figure 18.23 (a) The result of combining two opposite-facing soft threshold functions to produce a ridge. (b) The result of combining two ridges to produce a bump.

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

Supervised Learning: Summary So Far

learning as function approximation

k-NN: distance function (any attributes), any labels
Neural network: numeric attributes, numeric or binary labels

Regression: incremental training with LMS
3-Layer ANN: train with BackProp

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!