Neural Networks

(Chapter 9)

Kai Goebel, Bill Cheetham
GE Corporate Research & Development
goebel@cs.rpi.edu
cheetham@cs.rpi.edu

Stuff we talked about

Categories
- Unsupervised Learning
- Reinforcement Learning
- Unsupervised Learning

Hopfield Net
Perceptron
- Single Layer
- Multi Layer
Radial Basis Function Network

Neurons

Real neuron

Artificial neuron

Hopfield Network Operation

Picture is pattern; stored as attractor in the configuration space.
From arbitrary starting points, one attractor will be found

Hopfield Example

Learn $x=\begin{bmatrix} 1 & -1 & -1 \end{bmatrix}$
which gives us the weight matrix

$w=\begin{bmatrix} 0 & 1 & -1 \\
1 & 0 & -1 \\
-1 & -1 & 0 \end{bmatrix}$

Now let’s check the slightly corrupted pattern
$p=\begin{bmatrix} 1 & 1 & -1 \end{bmatrix}$
which will restore the pattern found close
$y=\begin{bmatrix} 1 & 1 & -1 \end{bmatrix}$
with an energy level of $E=-6$
More Complex Hopfield Examples

Reconstruction of Images

binary images are 130x180 pixels

Single-Layer Perceptron

Example: Gender classification (according to Jang)

Network Arch.

Training data

Learning:

select an input vector
if the response is incorrect, modify all weights

Two-Layer Perceptron: XOR

Node output as surface of their two inputs

note location of "o" and "x"

Multilayer Perceptrons (MLPs)

Network architecture

Learning rule:

- Steepest descent (Backprop)
- Conjugate gradient method
- All optim. methods using first derivative
- Derivative-free optim.

Activation function

Back-prop Neural Networks

Make incremental change in the direction dE/dw to decrease the error.
The learning rule for each node can be derived using the chain rule...

...to propagate the error back through a multi-layer perceptron 1. Initialize weights to small random values
2. Choose a pattern and apply it to input layer
3. Propagate the signal forward through the network
4. Compute the deltas for the output layer
5. Compute the deltas for the preceding layers by propagating the error backwards
6. Update all weights
7. Go back to step 2 and repeat for next pattern
8. Repeat until error rate is acceptable

Momentum

If error minimum in long narrow valley, then updating can happen to zig-zag down the valley

smoothes weight updating
can speed learning up

Page 2
Local Minima

There is no guarantee that the algorithm converges to a global minimum

- check with different initial conditions (different weights, etc.)
- perturb the system (data) with noise to improve result

MLP Decision Boundaries

Radial Basis Function (RBF) Networks

Network architecture

Each node is described by a bell shaped function

\[ a_i = \exp \left( -\frac{(x - c_i)^2}{\sigma^2} \right) \]

where

- \( c_i \) is the center of the curve
- Output: weighted sum
- linear combination

RBF and FIS

Consider the radial basis functions:

\[ y = \sum_i a_i \]

and a linear combination of the output variables

\[ y = \sum_i a_i \phi \]

then the response is equivalent to ...

XOR, revisited