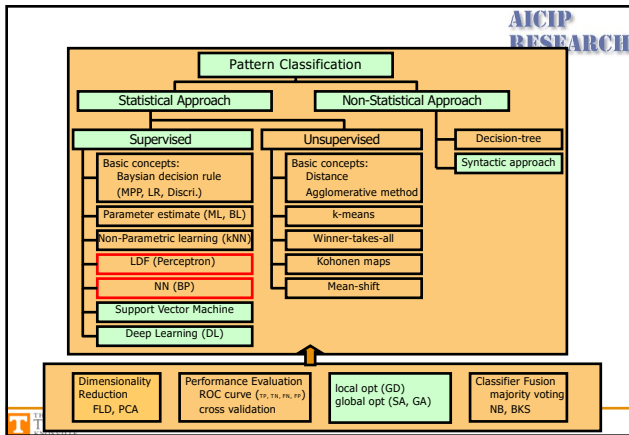
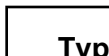



ECE471-571 – Pattern Recognition

Lecture 16: NN – Back Propagation


Hairong Qi, Gonzalez Family Professor
 Electrical Engineering and Computer Science
 University of Tennessee, Knoxville
<http://www.eecs.utk.edu/faculty/qi>
 Email: hqi@utk.edu



Types of NN

- ◆ Recurrent (feedback during operation)
 - Hopfield
 - Kohonen
 - Associative memory
- ◆ Feedforward
 - No feedback during operation or testing (only during determination of weights or training)
 - Perceptron
 - **Multilayer perceptron and backpropagation**

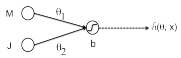
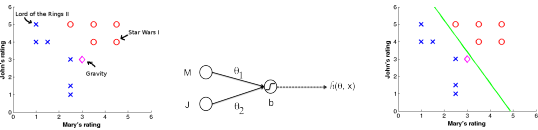

3

Limitations of Perceptron

- The output only has two values (1 or 0)
- Can only classify samples which are linearly separable (straight line or straight plane)
- Single layer: can only train AND, OR, NOT
- Can't train a network functions like XOR

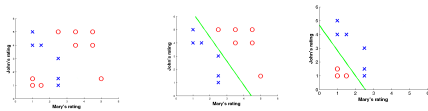
Why Deeper?

Movie name	Mary's rating	John's rating	I like?
Lord of the Rings II	1	5	No
...
Star Wars I	4.5	4	Yes
Gravity	3	3	?

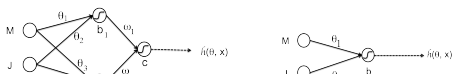


<http://ai.stanford.edu/~quocle/tutorial2.pdf>

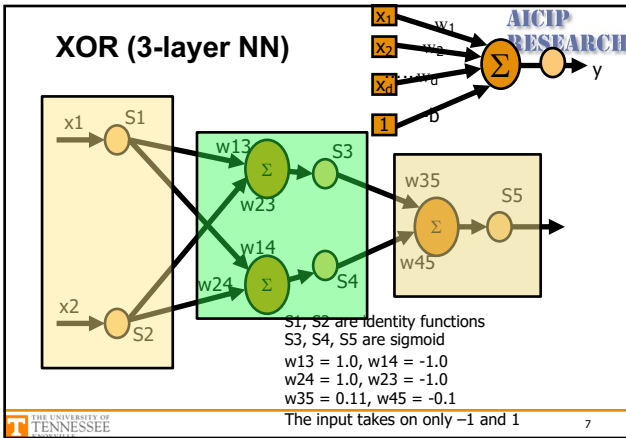
Why Deeper? (Cont')

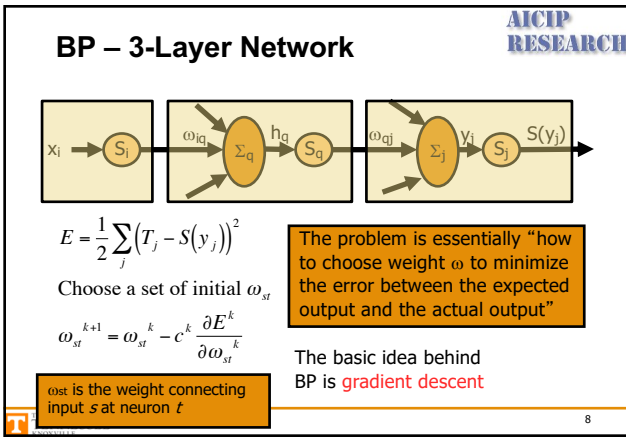


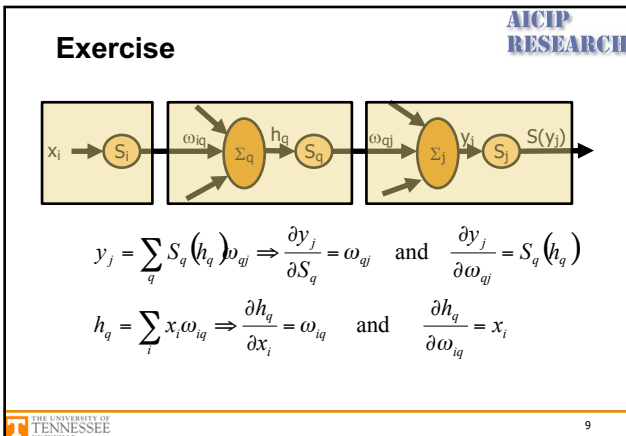
Movie name	Output by decision function h_1	Output by decision function h_2	Simon likes?
Lord of the Rings II	$h_1(p^{(1)})$	$h_2(p^{(1)})$	No
Star Wars I	$h_1(p^{(2)})$	$h_2(p^{(2)})$	Yes
Gravity	$h_1(p^{(3)})$	$h_2(p^{(3)})$?



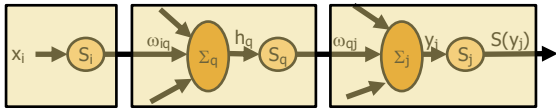
<http://ai.stanford.edu/~quocle/tutorial2.pdf>







*The Derivative – Chain Rule



$$\Delta\omega_{aj} = -\frac{\partial E}{\partial \omega_{aj}} = -\frac{\partial E}{\partial S_j} \frac{\partial S_j}{\partial y_j} \frac{\partial y_j}{\partial \omega_{aj}}$$

$$= -(T_j - S_j)(S'_j)(S'_q)(h_q)$$

$$\Delta\omega_{iq} = -\frac{\partial E}{\partial \omega_{iq}} = \left[\sum_j \frac{\partial E}{\partial S_j} \frac{\partial S_j}{\partial y_j} \frac{\partial y_j}{\partial S_q} \right] \frac{\partial S_q}{\partial h_q} \frac{\partial h_q}{\partial \omega_{iq}}$$

$$= \left[\sum_j (T_j - S_j)(S'_j)(\omega_{aj}) \right] (S'_q)'(x_i)$$

Threshold Function

- ◆ Traditional threshold function as proposed by McCulloch-Pitts is binary function
- ◆ The importance of differentiable
- ◆ A threshold-like but differentiable form for S (25 years)
- ◆ The sigmoid

$$S(x) = \frac{1}{1 + \exp(-x)}$$

*BP vs. MPP

$$E(\omega) = \sum_x [g_k(\mathbf{x}; \mathbf{w}) - T_k]^2 = \sum_{x \in \omega_k} [g_k(\mathbf{x}; \mathbf{w}) - 1]^2 + \sum_{x \notin \omega_k} [g_k(\mathbf{x}; \mathbf{w}) - 0]^2$$

$$= n \left\{ \frac{n_k}{n} \frac{1}{n_k} \sum_{x \in \omega_k} [g_k(\mathbf{x}; \mathbf{w}) - 1]^2 + \frac{n - n_k}{n} \frac{1}{n - n_k} \sum_{x \notin \omega_k} [g_k(\mathbf{x}; \mathbf{w})]^2 \right\}$$

$$\lim_{n \rightarrow \infty} \frac{1}{n} E(\mathbf{w}) = P(\omega_k) \int [g_k(\mathbf{x}; \mathbf{w}) - 1]^2 p(\mathbf{x} | \omega_k) d\mathbf{x} + P(\omega_{\neq k}) \int g_k^2(\mathbf{x}; \mathbf{w}) p(\mathbf{x} | \omega_{\neq k}) d\mathbf{x}$$

$$= \int [g_k^2(\mathbf{x}; \mathbf{w}) - 2g_k(\mathbf{x}; \mathbf{w}) + 1] p(\mathbf{x}, \omega_k) d\mathbf{x} + \int g_k^2(\mathbf{x}; \mathbf{w}) p(\mathbf{x}, \omega_{\neq k}) d\mathbf{x}$$

$$= \int g_k^2(\mathbf{x}; \mathbf{w}) p(\mathbf{x}) d\mathbf{x} - 2 \int g_k(\mathbf{x}; \mathbf{w}) p(\mathbf{x}, \omega_k) d\mathbf{x} + \int p(\mathbf{x}, \omega_k) d\mathbf{x}$$

$$= \int [g_k(\mathbf{x}; \mathbf{w}) - P(\omega_k | \mathbf{x})]^2 p(\mathbf{x}) d\mathbf{x} + C$$

THE UNIVERSITY OF TENNESSEE KNOXVILLE AICIP RESEARCH

Practical Improvements to Backpropagation

THE UNIVERSITY OF TENNESSEE KNOXVILLE AICIP RESEARCH

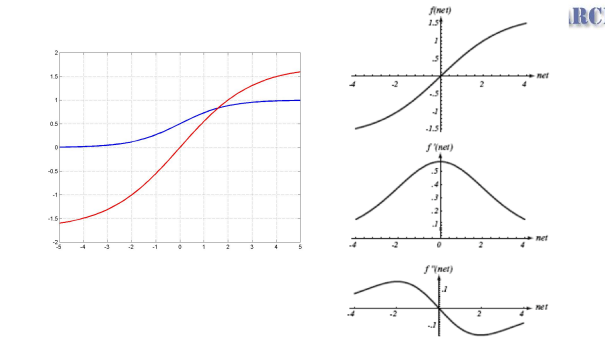
Activation (Threshold) Function

- The signum function

$$S(x) = \text{signum}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$
- The sigmoid function
 - Nonlinear
 - Saturate
 - Continuity and smoothness
 - Monotonicity (so $S'(x) > 0$)
$$S(x) = \text{sigmoid}(x) = \frac{1}{1 + \exp(-x)}$$
- Improved
 - Centered at zero
 - Antisymmetric (odd) – leads to faster learning
 - $a = 1.716, b = 2/3$, to keep $S'(0) \rightarrow 1$, the linear range is $-1 < x < 1$, and the extrema of $S''(x)$ occur roughly at $x \rightarrow 2$
$$S(x) = \text{sigmoid}(x) = \frac{2a}{1 + \exp(-bx)} - a$$

THE UNIVERSITY OF TENNESSEE KNOXVILLE 14

THE UNIVERSITY OF TENNESSEE KNOXVILLE AICIP RESEARCH



THE UNIVERSITY OF TENNESSEE KNOXVILLE 15

Data Standardization

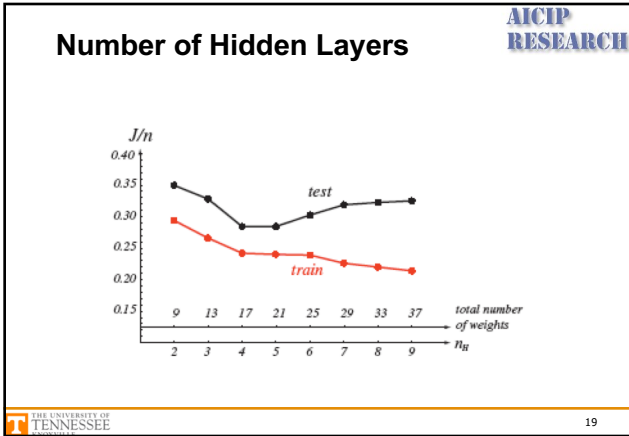
- Problem in the units of the inputs
 - Different units cause magnitude of difference
 - Same units cause magnitude of difference
- Standardization – scaling input
 - Shift the input pattern
 - The average over the training set of each feature is zero
 - Scale the full data set
 - Have the same variance in each feature component (around 1.0)

Target Values (output)

- ◆ Instead of one-of-c (c is the number of classes), we use +1/-1
 - +1 indicates target category
 - -1 indicates non-target category
- ◆ For faster convergence

Number of Hidden Layers

- The number of hidden layers governs the expressive power of the network, and also the complexity of the decision boundary
- More hidden layers -> higher expressive power -> better tuned to the particular training set -> poor performance on the testing set
- Rule of thumb
 - Choose the number of weights to be roughly $n/10$, where n is the total number of samples in the training set
 - Start with a “large” number of hidden units, and “decay”, prune, or eliminate weights



- ### Initializing Weight
- AICIP RESEARCH**
- Can't start with zero
 - Fast and uniform learning
 - All weights reach their final equilibrium values at about the **same time**
 - Choose weights randomly from a **uniform distribution** to help ensure uniform learning
 - **Equal negative and positive** weights
 - Set the weights such that the integration value at a hidden unit is in the range of **-1 and +1**
 - Input-to-hidden weights: $(-1/\sqrt{d}, 1/\sqrt{d})$
 - Hidden-to-output weights: $(-1/\sqrt{n_H}, 1/\sqrt{n_H})$, n_H is the number of connected units
- THE UNIVERSITY OF TENNESSEE**
- 20

Learning Rate

AICIP RESEARCH

$$c_{opt} = \left(\frac{\partial^2 MSE}{\partial \omega^2} \right)^{-1}$$

- ◆ The optimal learning rate
 - Calculate the 2nd derivative of the objective function with respect to each weight
 - Set the optimal learning rate separately for each weight
 - A learning rate of **0.1** is often adequate

THE UNIVERSITY OF TENNESSEE

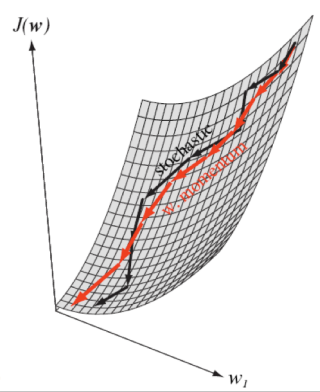
21

Plateaus or Flat Surface in S'

- ◆ Plateaus
 - Regions where the derivative $\frac{\partial E}{\partial \omega}$ is very small
 - When the sigmoid function saturates
- ◆ Momentum
 - Allows the network to learn more quickly when plateaus in the error surface exist

$$\omega_{st}^{k+1} = \omega_{st}^k - c^k \frac{\partial E^k}{\partial \omega_{st}^k}$$

$$\omega_{st}^{k+1} = \omega_{st}^k + (1 - c^k) \Delta \omega_{pp}^k + c^k (\omega_{st}^k - \omega_{st}^{k-1})$$



Weight Decay

- ◆ Should almost always lead to improved performance

$$\omega^{new} = \omega^{old} (1 - \epsilon)$$

**AICIP
RESEARCH**

Batch Training vs. On-line Training

- ◆ Batch training
 - Add up the weight changes for all the training patterns and apply them in one go
 - GD
- ◆ On-line training
 - Update all the weights immediately after processing each training pattern
 - Not true GD but faster learning rate

THE UNIVERSITY OF TENNESSEE 25

**AICIP
RESEARCH**

Other Improvements

- ◆ Other error function (Minkowski error)

THE UNIVERSITY OF TENNESSEE 26

**AICIP
RESEARCH**

Further Discussions

- How to draw the decision boundary of BPNN?
- How to set the range of valid output
 - 0-0.5 and 0.5-1?
 - 0-0.2 and 0.8-1?
 - 0.1-0.2 and 0.8-0.9?
- The importance of having symmetric initial input

THE UNIVERSITY OF TENNESSEE 27
