

Soft Computing: Unsupervised Learning

Unsupervised Learning

(chapter 11)

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

1

Soft Computing: Unsupervised Learning

Stuff we will talk about

- Competitive Learning Networks**
- Kohonen Self-Organizing Networks**
- Learning Vector Quantization**

2

Soft Computing: Unsupervised Learning

Introduction

When no teacher or critic is available, only the input vectors can be used for learning.

Learning system categorizes or detects features without feedback

=> use for clustering, feature extraction, similarity detection, data mining

3

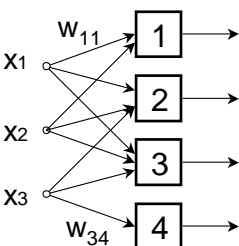
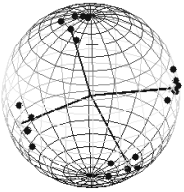
Soft Computing: Unsupervised Learning

Competitive Learning

Winner takes all:

Weights of the unit with highest activation are updated

The weight vector rotates slowly towards the cluster centers

4

Competitive Learning

Activation of output unit j

$$a_j = \sqrt{\sum_{i=1}^3 (x_i - w_{ij})^2} = \|x - w_j\|$$

and the weights are updated according to:

$$w_k(t+1) = w_k(t) + \eta(x(t) - w_k(t))$$

Note: different metrics can be used leading to different solutions

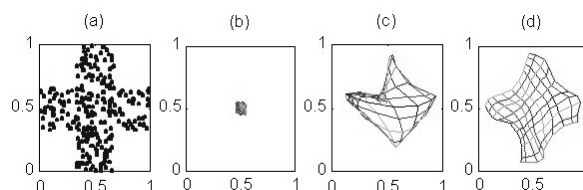
**If initial weights are far from actual centers:
may never get updated;**

=> use leaky learning

5

Self-Organizing Networks (Kohonen)

- Learning is similar to Competitive Learning
- Not only the winning units are updated but all the weights in a neighborhood
- Size of neighborhood decreases
- Topological properties in the input data is reflected in the output units through neighborhood constraints



6

	Soft Computing: Unsupervised Learning
	Self-Organizing Networks →
	<p>Algorithm</p> <p>- Step 1</p> <p>Select winning output (smallest dissimilarity)</p> $\ x - w_c\ = \min_i \ x - w_i\ $ <p>- Step 2</p> <p>Update winners and neighborhood NB_c of winner c</p> $\Delta w_i = \eta(x - w_i) \quad i \in NB_c$ <p>Reduce η gradually</p>
7	

	Soft Computing: Unsupervised Learning
	Learning Vector Quantization (LVQ) →
	<p>Adaptive data classification</p> <ol style="list-style-type: none"> 1. Cluster data (using any clustering tool) 2. Label data using majority of data in cluster "voting method" <p>Fine tune class information to minimize errors by</p> <ol style="list-style-type: none"> 3. Finding cluster center closest to input 4. If x and w_k belong to the same class: $\Delta w_k = \eta(x - w_k)$ <p>otherwise</p> $\Delta w_k = -\eta(x - w_k)$ <p>Repeat until max. number of iterations reached</p>
8	

	<p>Soft Computing: Unsupervised Learning</p> <h2>Adaptive Resonance Theory (ART)</h2>
9	<ul style="list-style-type: none"> - Accept and adapt stored prototypes of a class when the input is sufficiently similar to it. - Input and stored prototype are said to “resonate”. - If the input is not sufficiently similar to any class, then form a new class. - Similarity is judged by a “vigilance” level - ART1 focuses on binary input. - ART2 designed for continuous input.

	<p>Soft Computing: Unsupervised Learning</p> <h2>ART1</h2>
10	<p>Algorithm</p> <ol style="list-style-type: none"> 1. Enable all output units 2. Find the winner among all enabled output units by comparing the components of the input vector. 3. Check whether the match is good enough by comparing the ratio of bits in input and prototype to vigilance level 4. If the match is good, adjust winning vector by adjusting any bits that are not also in input vector. 5. If the match is not good enough, create a new class by adding the current vector as a class prototype.

Soft Computing: Unsupervised Learning

last slide

11

The slide frame consists of a rectangular border. At the top, there is a title bar containing the text 'Soft Computing: Unsupervised Learning'. A horizontal arrow points from the left side of the frame towards the right. In the center of the frame, the text 'last slide' is displayed. In the bottom-left corner, the number '11' is present. There is a small square in the top-right corner and a dotted arc in the top-left corner.