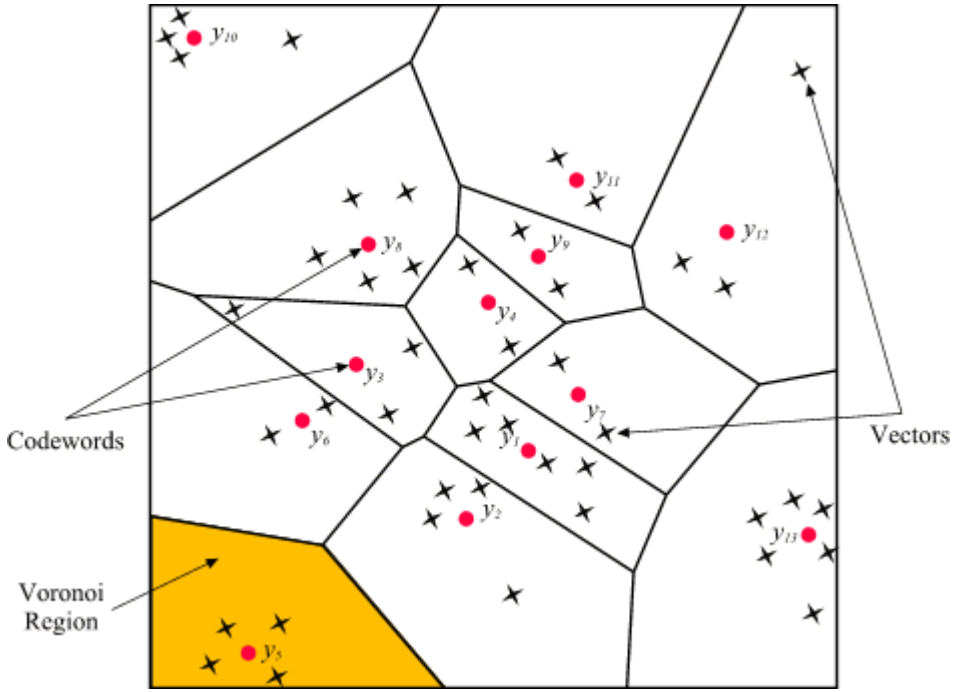
In The Name of Allah

Neural Networks

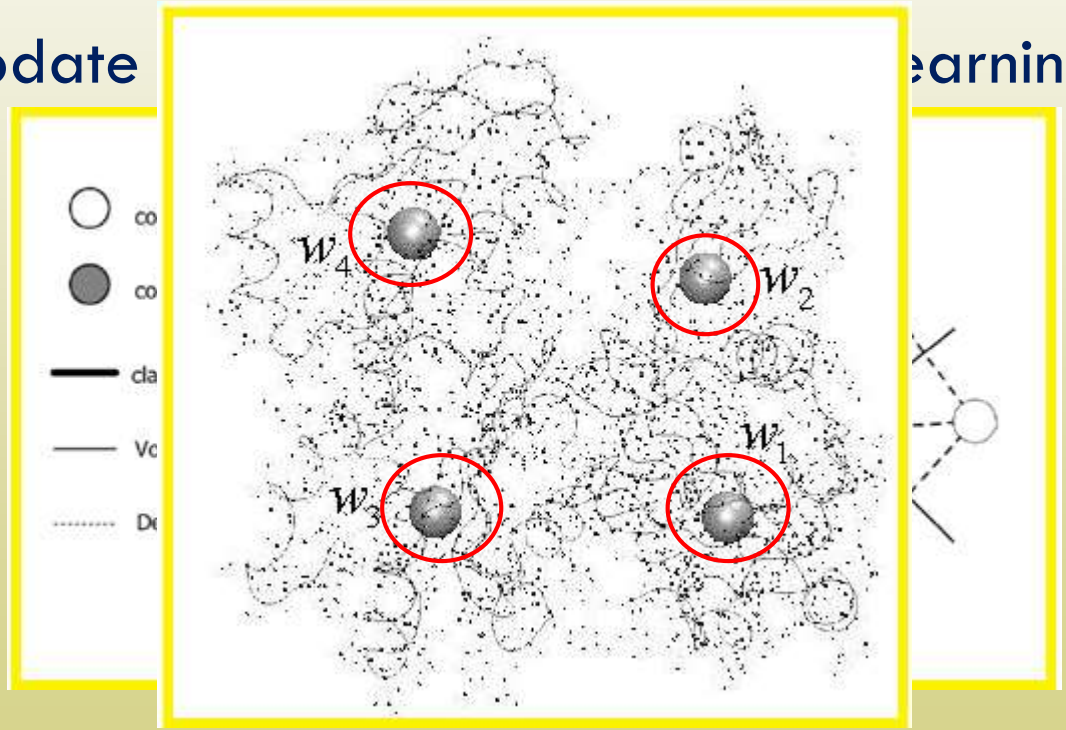
Learning Vector Quantization



Shahrood University of Technology
Hossein Khosravi

Introduction

- General idea
  - Assign some prototypes for each class
  - Update earning



Learning Vector Quantization (LVQ)

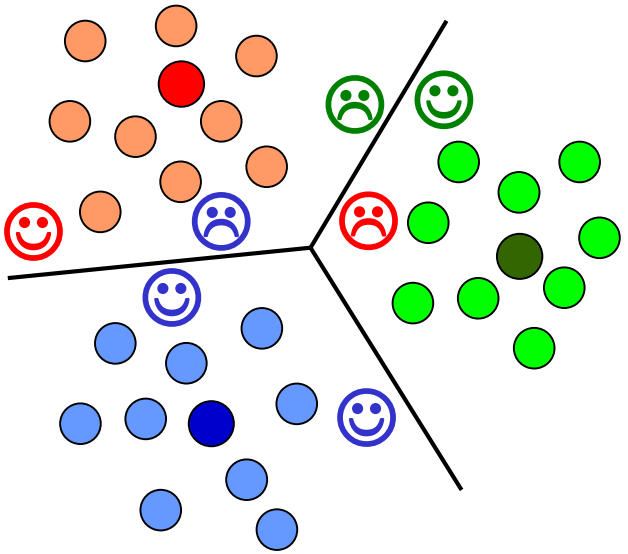
aim:

classification of data

learning from examples

example situtation:

3 classes , 3 prototypes



classification:

assignment of a vector ξ

to the class of the closest

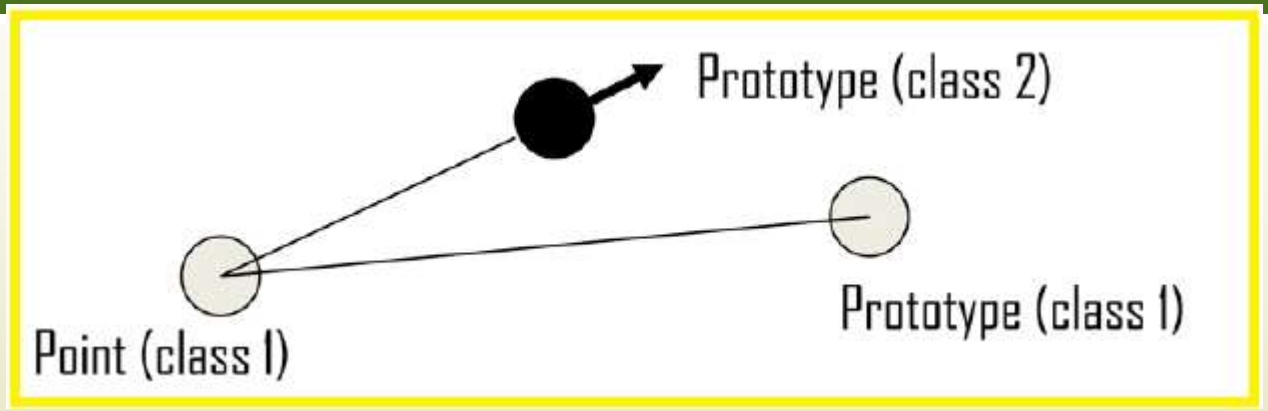
prototype **w**

Learning: choice of prototypes according to example data

aim : **generalization ability**, i.e. correct classification of *novel* data after training

LVQ Algorithms

- LVQ 1



$$c = argmin(||x^q - w^m||)$$

$$w^c(t+1) = w^c(t) + a(t)s(t)[x^q - w^c(t)]$$

$$0 < a(t) < 1$$

$$s(t) = \begin{cases} +1 & if\ classification\ is\ correct \\ -1 & if\ classification\ is\ wrong \end{cases}$$

LVQ 2.1

- Concurrent update of the two nearest prototypes

- Three conditions must be meet for update:
  - Nearest prototype (m_i) is from wrong class
  - The second nearest one (m_j), is from the correct class
  - The input vector is sufficiently close to the decision boundary between m_i and m_j:

0.4<S<0.8 $\quad\quad$ $\min(\frac{d_i}{d_j}, \frac{d_j}{d_i}) \geq S$

d_i , d_j: distance between input and prototype

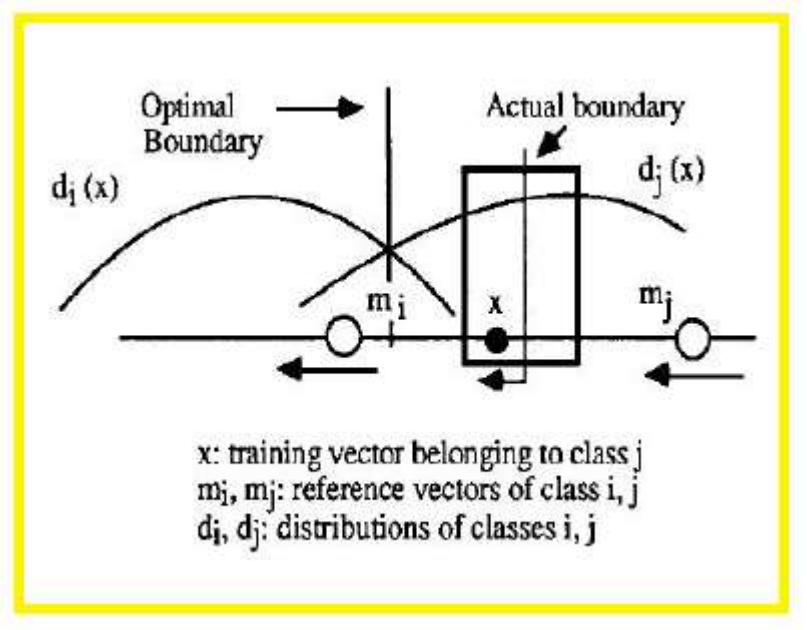
- $m_j(t+1) = m_j(t) + a(t)[x(t) - m_j(t)]$

  $m_i(t+1) =$ $(t)]$

- $\cdot \frac{i}{N}\Big)$ i

  $= 1, \ldots, N$

- $a(0) = 0.08$



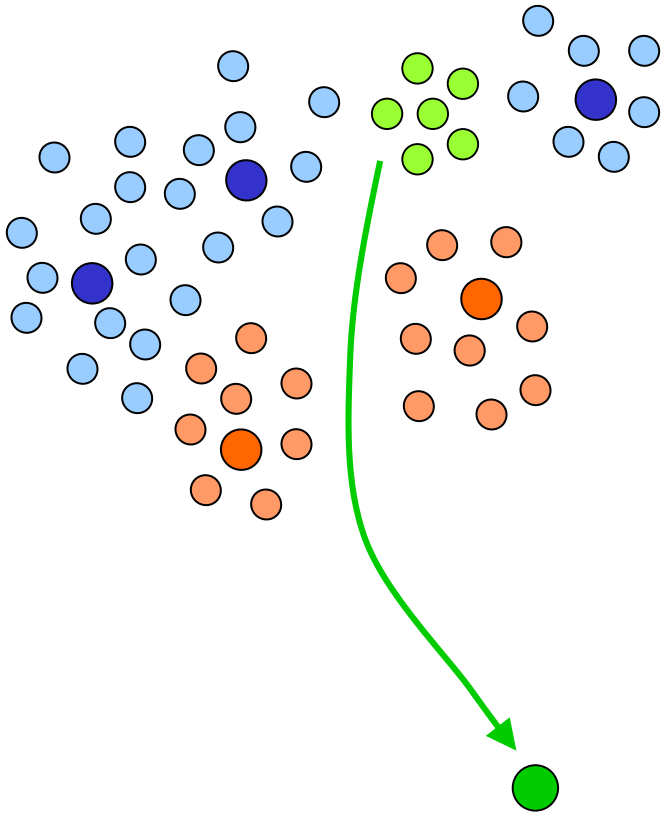
mostly: **heuristically** motivated variations of **competitive learning**

prominent example [Kohonen]: **“ LVQ 2.1. ”**

- initialize prototype vectors (for different classes)

- present a single example
- identify the closest *correct* and the closest *wrong* prototype
- move the corresponding *winner* towards / away from the example

known convergence / stability problems,
e.g. for infrequent classes



LVQ As A Neural Net

- One hidden layer can be assumed