HW #1. Fuzzy ART Network


- Using the skeleton codes and referring to the procedure on p.3 of this ppt, fill the following 4 functions in ‘Fill’ folder:
  - complementCoding.m, activateART.m, matchART.m, and updateART.m

- In the report, discuss the following issues:
  - Deeper searches of previously coded categories with initial weights $w_{ji} > 1$.
  - The classification performance w.r.t. $\alpha$, $\beta$, and $\rho$
    - # of categories
    - Fast-commit slow-recode option
    - Conservative limit by taking $\alpha \rightarrow 0$ to minimize recoding during learning
    - The performance for one-shot stable learning
• Complement coding for pattern generalization while avoiding the category proliferation problem
  - Note that pattern generalization is explained as follows:
    Input $I=(1, 0)$: distance 1 btw $I_1$ and $I_2$
    $\rightarrow$ Complement coded input $I=\{(1, 0), (0, 1)\}$: distance $\sqrt{2}$ btw $I_1$ and $I_2$
    More discriminative for input patterns

• Strength and weakness of ART in comparison with K-Nearest Neighbors (KNN) and Support Vector Machine (SVM).

- Design your own ART network as a variant of the Fuzzy ART

- Submit the report along with main.m and the 4 programmed functions by zip file name: HW1_yourname.zip

- Due date: March 26, 2017

- Send to: yhyoo@rit.kaist.ac.kr
Fuzzy ART procedure

- Complement coding (*complementCoding.m*)
  - Let \( I = (I_1, I_2, \ldots, I_n) \) denote an input vector
  - Let \( x = (I, \bar{I}) \) be the activity vector (\( \bar{I} = 1 - I \))

- Code activation (*activateART.m*)
  - \( T_j = \frac{|x \land w_j|}{\alpha + |w_j|} \), where \( \alpha \) is a choice parameter, \( w_j \) is a weight vector that is linked to \( y_j \)

- Code competition
  - \( T_j = \max \{ T_j : \text{for all } F_2 \text{ node } j \} \)

- Template matching (*MatchART.m*)
  - \( m_j = \frac{|x \land w_j|}{|x|} \geq \rho \), where \( \rho \) is a vigilance parameter

- Template learning (*updateART.m*)
  - Select all \( i \) that satisfy \( x_i < w_{ij} \)
  - \( w_{ij} = \beta x_i + (1 - \beta)w_{ij} \)
HW #1. Fuzzy ART Network

Results

- **Original Input**

- **Learning rate: 0.5, vigilance: 0.2**

- **Learning rate: 0.5, vigilance: 0.5**

- **Learning rate: 0.5, vigilance: 0.8**
K-Nearest Neighbors (KNN)

- KNN stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions)

- Requires three things
  - The set of stored records
  - Distance metric to compute distance between records
  - The value of \( k \), the number of nearest neighbors to retrieve

- To classify an unknown record
  - Compute distance to other training records
  - Identify \( k \) nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record, e.g., by taking majority vote

In case of \( k=1 \): Blue, \( k=3 \): Red, \( k=5 \): Blue
Support Vector Machine (SVM)

- Linear classifier
  - There are a lot of possible solutions for weight $w$ and bias $b$ of decision boundary
  - Some methods find a separating hyperplane, but not the optimal one according to some criterion of expected goodness, e.g., perceptron

- SVM finds an optimal solution
  - Maximizes the distance between the hyperplane and the ‘support points’ close to decision boundary
  - One intuition:
    If no points near the decision surface, then no very uncertain classification decisions

Decision boundary: $w_1x + w_2y + b = 0$
Support Vector Machine (SVM)

- SVMs maximize the margin around the separating hyperplane as known as large margin classifiers.
- The decision function is fully specified by a subset of training samples, the support vectors on maximum margin hyperplane.
- Solving SVMs is a quadratic programming problem.

- Find $w$ and $b$ such that $\rho = \frac{2}{\|w\|}$ is maximized.
  
  - For all $\{(x_i, y_i)\}$ where $y_i \in \{+1, -1\}$,
    
    $w^T x_i + b \geq 1$, if $y_i = 1$
    $w^T x_i + b \leq -1$, if $y_i = -1$