Artificial Neural Networks

LVQ, Hopfield, ART, and ACE

Homework #4

Poorya MohammadiNasab

(400722138)

Contents

Problem 1

Assume we have the following patterns that look like English alphabets. We want to train an ART1 model that can discriminate between them. Determine your parameters and then train the model on the training set. Then, introduce the test sample to the network and classify it. Can your model perform well on continuous patterns? Why? (35 pts)

Considerations:

1. Choose your training and test set according to your student id.
2. Train your model with an arbitrary vigilance.
3. Train the model until convergence.
4. You don't have to write code, but you can utilize it for ease of calculations.
5. Don't forget to define the F1 and F2 units of your network.
6. Specify the following items after introducing each pattern:
    - Weights
    - Distances
    - Active prototypes
    - Winner prototype
    - Similarity
    - Classes and their patterns

8 % 3 = 2:

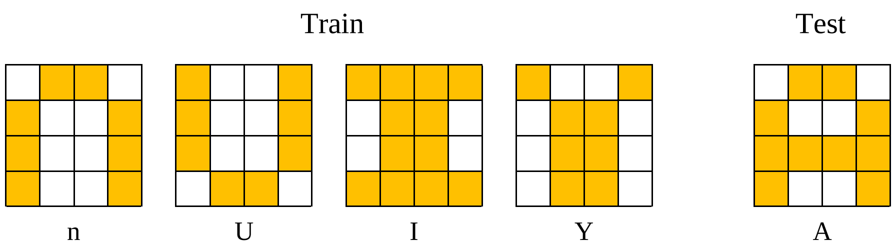
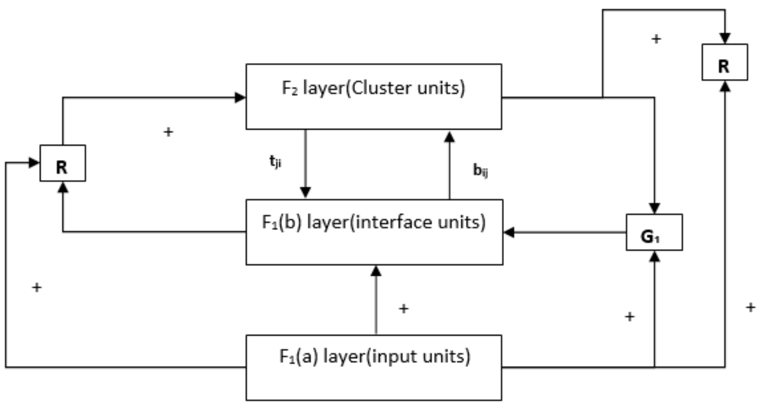


Figure 1 illustrate the ART model architecture.



The term “resonance” in ART is the state of the network, when a class of a prototype vector very closely matches the current input vector, leads to a state which permits learning. During this resonant state, the weight updation takes place.

Vigilance parameter (ρ) = 0.4

Learning rate (α) = 2

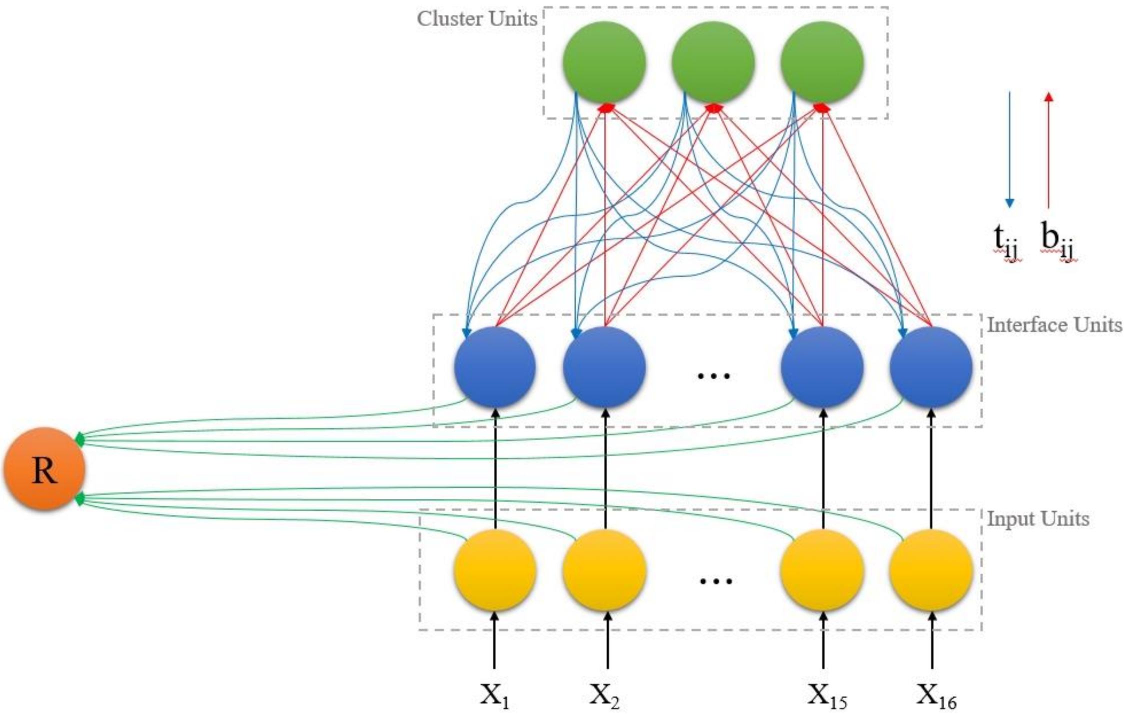
Number of features (n) = 16

Number of clusters (m) = 3

Input vectors:

n	0	1	1	0	1	0	0	1	1	0	0	1	1	0	0	1
U	1	0	0	1	1	0	0	1	1	0	0	1	0	1	1	0
I	1	1	1	1	0	1	1	0	0	1	1	0	1	1	1	1
Y	1	0	0	1	0	1	1	0	0	1	1	0	0	1	1	0

Network architecture for this problem:



Bottom-up-weights, $b_{ij}(0) = \frac{1}{1+n} = \frac{1}{17} = 0.06$

Top-down-weights, $t_{ij}(0) = 1$

$$b_{ij} = \begin{bmatrix} 0.06 & 0.06 & 0.06 \\ \vdots & \vdots & \vdots \\ 0.06 & 0.06 & 0.06 \end{bmatrix}_{16\times 3}$$

$$t_{ij} = \begin{bmatrix} 1 & \cdots & 1 \\ 1 & \cdots & 1 \\ 1 & \cdots & 1 \end{bmatrix}_{3\times 16}$$

First input

Compute norm of "n": ||n|| = sum (n) = 8

$$y_i = \sum_{i=1}^{4} b_{ij} x_i$$

x = [0110100110011001]

$$y_1 = 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) = 0.48$$

$$y_2 = 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) = 0.48$$

$$y_3 = 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) = 0.48$$

Winner cluster = 1

$$x_i = s_i\, t_{ji}$$

$$x_1 = [0110100110011001][1111111111111111] = [0110100110011001]$$

Computer norm of "x_1" = sum (x_1) = 8

Test for reset condition: $\frac{||x||}{||s||} = \frac{8}{8} = 1.0 \geq 0.4$ (ρ)

Reset is false.

Update bottom-up-weights for α = 2

$$b_{ij}(new) = \frac{\alpha x_i}{\alpha - 1 + ||x||}$$

$b_{11} = \dfrac{2\times 0}{1+8} = 0,\quad b_{21} = \dfrac{2\times 1}{1+8} = 0.2,\quad b_{31} = \dfrac{2\times 1}{1+8} = 0.2,\quad b_{41} = \dfrac{2\times 0}{1+8} = 0$

$b_{51} = \dfrac{2\times 1}{1+8} = 0.2,\quad b_{61} = \dfrac{2\times 0}{1+8} = 0,\quad b_{71} = \dfrac{2\times 0}{1+8} = 0,\quad b_{81} = \dfrac{2\times 1}{1+8} = 0.2$

$b_{91} = \dfrac{2\times 1}{1+8} = 0.2,\quad b_{10,1} = \dfrac{2\times 0}{1+8} = 0,\quad b_{11,1} = \dfrac{2\times 0}{1+8} = 0,\quad b_{12,1} = \dfrac{2\times 1}{1+8} = 0.2$

$b_{13,1} = \dfrac{2\times 1}{1+8} = 0.2,\quad b_{14,1} = \dfrac{2\times 0}{1+8} = 0,\quad b_{15,1} = \dfrac{2\times 0}{1+8} = 0,\quad b_{16,1} = \dfrac{2\times 1}{1+8} = 0.2$

$$b_{ij} = \begin{bmatrix} 0 & 0.06 & 0.06 \\ \vdots & \vdots & \vdots \\ 0.2 & 0.06 & 0.06 \end{bmatrix}_{16\times 3}$$

Update t_{ij}:

$$t_{ij} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & \cdots & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 \end{bmatrix}_{3\times 16}$$

Second input

Compute norm of "U": ||U|| = sum (U) = 8

$$y_i = \sum_{i=1}^{4} b_{ij} x_i$$

x = [1001100110010110]

$$y_1 = 0(1) + 0.2(0) + 0.2(0) + 0(1) + 0.2(1) + 0(0) + 0(0) + 0.2(1) + 0.2(1) + 0(0) + 0(0) \\ + 0.2(1) + 0.2(0) + 0(1) + 0(1) + 0.2(0) = 0.8$$

$$y_2 = 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) \\ + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) = 0.48$$

$$y_3 = 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) \\ + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) = 0.48$$

Winner cluster = 2

$$x_i = s_i\, t_{ji}$$

$$x_2 = [1001100110010110][1111111111111111] = [1001100110010110]$$

Computer norm of "x_2" = sum (x_2) = 8

Test for reset condition: $\frac{||x||}{||s||} = \frac{8}{8} = 1.0 \geq 0.4$ (ρ)

Reset is false.

Update bottom-up-weights for α = 2

$$b_{ij}(new) = \frac{\alpha x_i}{\alpha - 1 + ||x||}$$

$$b_{12} = \frac{2\times 1}{1+8} = 0.2, \quad b_{22} = \frac{2\times 0}{1+8} = 0, \quad b_{32} = \frac{2\times 0}{1+8} = 0, \quad b_{42} = \frac{2\times 1}{1+8} = 0.2$$

$$b_{52} = \frac{2\times 1}{1+8} = 0.2, \quad b_{62} = \frac{2\times 0}{1+8} = 0, \quad b_{72} = \frac{2\times 0}{1+8} = 0, \quad b_{82} = \frac{2\times 1}{1+8} = 0.2$$

$$b_{92} = \frac{2\times 1}{1+8} = 0.2, \quad b_{10,2} = \frac{2\times 0}{1+8} = 0, \quad b_{11,2} = \frac{2\times 0}{1+8} = 0, \quad b_{12,2} = \frac{2\times 1}{1+8} = 0.2$$

$$b_{13,2} = \frac{2\times 0}{1+8} = 0, \quad b_{14,2} = \frac{2\times 1}{1+8} = 0.2, \quad b_{15,2} = \frac{2\times 1}{1+8} = 0.2, \quad b_{16,2} = \frac{2\times 0}{1+8} = 0$$

$$b_{ij} = \begin{bmatrix} 0 & 0.2 & 0.06 \\ \vdots & \vdots & \vdots \\ 0.2 & 0 & 0.06 \end{bmatrix}_{16\times3}$$

Update t_{ij}:

$$t_{ij} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & \cdots & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 \end{bmatrix}_{3\times16}$$

Third input

Compute norm of "I": ||I|| = sum (I) = 12

$$y_i = \sum_{i=1}^{4} b_{ij} x_i$$

x = [1111011001101111]

$$y_1 = 0(1) + 0.2(1) + 0.2(1) + 0(1) + 0.2(0) + 0(1) + 0(1) + 0.2(0) + 0.2(0) + 0(1) + 0(1) + 0.2(0) + 0.2(1) + 0(1) + 0(1) + 0.2(1) = 0.8$$

$$y_2 = 0.2(1) + 0(1) + 0(1) + 0.2(1) + 0.2(0) + 0(1) + 0(1) + 0.2(0) + 0.2(0) + 0(1) + 0(1) + 0.2(0) + 0(1) + 0.2(1) + 0.2(1) + 0(1) = 0.8$$

$$y_3 = 0.06(1) + 0.06(1) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(0) + 0.06(1) + 0.06(1) + 0.06(1) + 0.06(1) = 0.72$$

Winner cluster = 3

$$x_i = s_i \, t_{ji}$$

$$x_3 = [1111011001101111][1111111111111111] = [1111011001101111]$$

Computer norm of "x_3" = sum (x_3) = 12

Test for reset condition: $\frac{||x||}{||s||} = \frac{8}{8} = 1.0 \geq 0.4\ (\rho)$

Reset is false.

Update bottom-up-weights for α = 2

$$b_{ij}(new) = \frac{\alpha x_i}{\alpha - 1 + ||x||}$$

$b_{13} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{23} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{33} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{43} = \dfrac{2\times1}{1+12} = 0.15$

$b_{53} = \dfrac{2\times0}{1+12} = 0,\quad b_{63} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{73} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{83} = \dfrac{2\times0}{1+12} = 0$

$b_{93} = \dfrac{2\times0}{1+12} = 0,\quad b_{10,3} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{11,3} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{12,3} = \dfrac{2\times0}{1+12} = 0$

$b_{13,3} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{14,3} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{15,3} = \dfrac{2\times1}{1+12} = 0.15,\quad b_{16,3} = \dfrac{2\times1}{1+12} = 0.15$

$$b_{ij} = \begin{bmatrix} 0 & 0.2 & 0.15 \\ \vdots & \vdots & \vdots \\ 0.2 & 0 & 0.15 \end{bmatrix}_{16\times 3}$$

Update t_{ij}:

$$t_{ij} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & \cdots & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & \cdots & 1 \end{bmatrix}_{3\times 16}$$

Forth input

Compute norm of “Y”: ||Y|| = sum (Y) = 8

$$y_i = \sum_{i=1}^{4} b_{ij} x_i$$

x = [1001011001100110]

$$y_1 = 0(1) + 0.2(0) + 0.2(0) + 0(1) + 0.2(0) + 0(1) + 0(1) + 0.2(0) + 0.2(0) + 0(1) + 0(1) + 0.2(0) + 0.2(0) + 0(1) + 0(1) + 0.2(0) = 0$$

$$y_2 = 0.2(1) + 0(0) + 0(0) + 0.2(1) + 0.2(0) + 0(1) + 0(1) + 0.2(0) + 0.2(0) + 0(1) + 0(1) + 0.2(0) + 0(0) + 0.2(1) + 0.2(1) + 0(0) = 0.8$$

$$y_3 = 0.15(1) + 0.15(0) + 0.15(0) + 0.15(1) + 0(0) + 0.15(1) + 0.15(1) + 0(0) + 0(0) + 0.15(1) + 0.15(1) + 0(0) + 0.15(0) + 0.15(1) + 0.15(1) + 0.15(0) = 1.2$$

Winner cluster = 3

$$x_i = s_i \, t_{ji}$$

$$x_1 = [1001011001100110][0110100110011001] = [0000000000000000]$$

Computer norm of “x_1” = sum (x_1) = 0

Test for reset condition: $\frac{||x||}{||s||} = \frac{0}{8} = 0.0 \leq 0.4$ (ρ)

Reset is True.

Test input

Compute norm of “A”: ||A|| = sum (A) = 10

$$y_i = \sum_{i=1}^{4} b_{ij} x_i$$

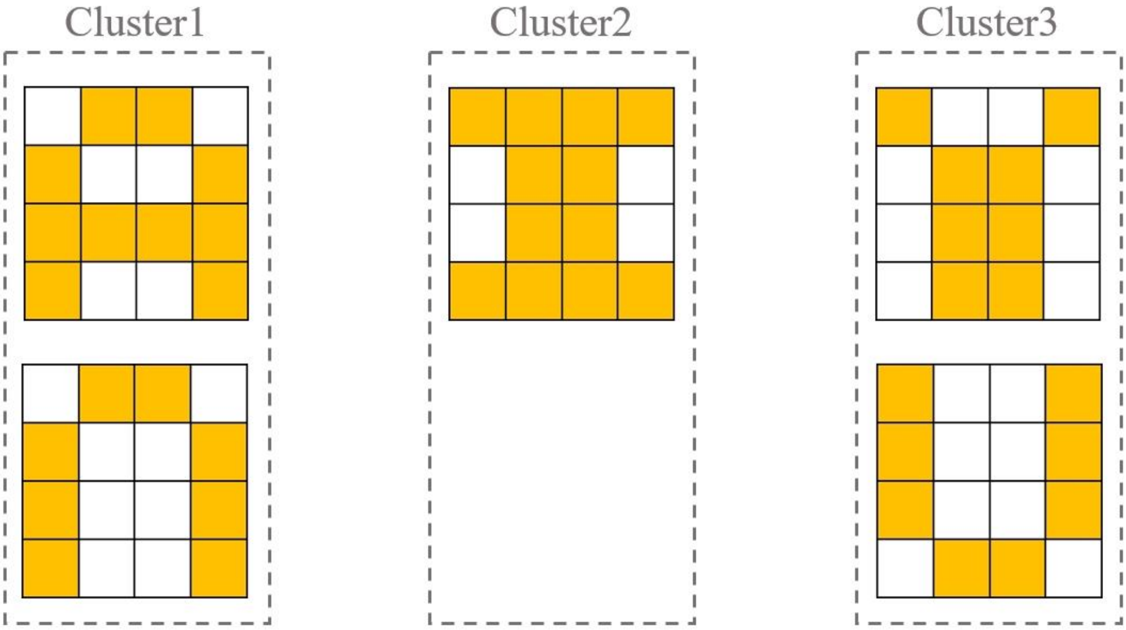
x = [0110100111111001]

$$y_1 = 0(0) + 0.2(1) + 0.2(1) + 0(0) + 0.2(1) + 0(0) + 0(0) + 0.2(1) + 0.2(1) + 0(1) + 0(1) + 0.2(1) + 0.2(1) + 0(0) + 0(0) + 0.2(1) = 1.6$$

$$y_2 = 0.2(0) + 0(1) + 0(1) + 0.2(0) + 0.2(1) + 0(0) + 0(0) + 0.2(1) + 0.2(1) + 0(1) + 0(1) + 0.2(1) + 0(1) + 0.2(0) + 0.2(0) + 0(1) = 0.8$$

$$y_3 = 0.15(0) + 0.15(1) + 0.15(1) + 0.15(0) + 0(1) + 0.15(0) + 0.15(0) + 0(1) + 0(1) + 0.15(1) + 0.15(1) + 0(1) + 0.15(1) + 0.15(0) + 0.15(0) + 0.15(1) = 0.9$$

Winner cluster = 1



It is impossible to use ART model for continuous samples, because its architecture is too simple and also it is a binary based algorithm. If we want to use continuous samples we can use ART2 model.

Problem2

Which of the following situations can occur after training an RCE network? Explain your yes or no answer in two lines at most. (15 pts)

a. Having concentric circles of the same class.

No, because in training process we cannot add a new circle for an existing class. So, it is impossible to have concentric circles.

b. Having concentric circles of different classes.

No, having a concentric circles of different classes means that two same samples, have two different classes and if we have this situation the bases of the algorithm are wrong.

c. Having tangent circles of the same class.

Yes, imagine we have two circles for two classes. The third point, does not belong to any of this two classes and we add a new circle. But, in fact one of the previous points are belongs to new circle. Generally, this situation depends on the order of points.

d. Having tangent circles of different classes.

Yes, it is a similar situation to the previous case (case “c”).

e. Having circles enclosed by another circle.

Yes and no, if circles belong to a same class it is possible. But, circles in different classes cannot enclosed by another circle.

Problem3

In this section, you should answer the following questions. (10 pts)

3 - A:

What is the drawback of the Hopfield network in comparison to the Boltzmann machine? How does Boltzmann overcome this problem?

Hopfield networks

- suffer from spurious local minima that form on the energy hyper-surface
- require the input patterns to be uncorrelated
- are limited in capacity of patterns that can be stored
- are usually fully connected and not stacked

Restricted Boltzmann Machines (RBMs) avoid the spurious solutions that arise in Hopfield Networks by adding in hidden nodes and then sampling over all possible nodes using Boltzmann statistics. This is not the only difference however, as older studies of Hopfield networks also looked at adding temperature (i.e. the little model for spin glasses) Spin Glass Models of Neural Networks.

3 - B:

Compare Hopfield and BSB model. Explain their applications separately.

Applications of BSB model:

- Associative memory design using BSB net
- Large scale BSB nets
- Storing and retrieving images

Applications of Hopfield model:

- Pattern Matching in Chart Analysis
- face recognition
- enhancement of X-Ray images
- medical image restoration

Comparing Hopfield and BSB model:

- The Hopfield network (model) consists of a set of neurons and a corresponding set of unit delays, forming a multiple-loop feedback system.
- The number of feedback loops is equal to the number of neurons.
- The output of each neuron is fed back via a unit delay element, to each of the other neurons in the network.
- An associative memory is a content-addressable structure that maps a set of input patterns to a set of output patterns.

References

1) www.quora.com/Why-use-reduced-Boltzmann-machines-instead-of-Hopfield-networks-for-deep-belief-networks

2) engineering.purdue.edu/~zak/Microsoft%20PowerPoint%20-%20Pelincec_slides.pdf\

3) www.mdpi.com/2076-3417/11/9/3876/pdf

4) researchgate.net/publication/308199526_Application_of_Hopfield_neural_network_for_face_recognition

5) www.quora.com/What-are-the-domains-application-of-Hopfield-neural-networks

6) towardsdatascience.com/learning-vector-quantization-ed825f8c807d

7) machinelearningmastery.com/implement-learning-vector-quantization-scratch-python/