Heaven's Light is Our Guide

Computer Science & Engineering

Rajshahi University of Engineering & Technology

Course No.: CSE 2102
Course Title: Sessional based on CSE 2101
Experiment No. 4
Name of the Experiment:

a. Design and implementation of Kohonen Self-organizing Neural Networks algorithm.
b. Design and implementation of Hopfield Neural Networks algorithm.

Course Outcomes: CO1
Learning Domain with Level: Cognitive (Applying, Analyzing, Evaluating & Creating)

4(a)

Kohonen Network Algorithm

1. Initialise network
Define $w_{ij}(t)\,(0 \le i \le n-1)$ to be the weight from input i to node j at time t. Initialise weights from the n inputs to the nodes to small random values. Set the initial radius of the neighbourhood around node j, $N_j(0)$, to be large.
2. Present input
Present input $x_0(t), x_1(t), x_2(t), \ldots, x_{n-1}(t)$, where $x_i(t)$ is the input to node i at time t.
3. Calculate distances
Compute the distance d_j between the input and each output node j, given by

$$d_j = \sum_{i=0}^{n-1} (x_i(t) - w_{ij}(t))^2$$

4. Select minimum distance
Designate the output node with minimum d_j to be $j*$.
5. Update weights
Update weights for node $j*$ and its neighbours, defined by the neighbourhood size $N_{j*}(t)$. New weights are

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)(x_i(t) - w_{ij}(t))$$

For j in $N_{j*}(t)$, $\quad 0 \le i \le n-1$

The term $\eta(t)$ is a gain term ($0 < \eta(t) < 1$) that decreases in time, so slowing the weight adaption. Notice that the neighbourhood $N_{j*}(t)$ decreases in size as time goes on, thus localising the area of maximum activity.

6. Repeat by going to 2.

Your task is to:

- **Apply and Design** a program for your dataset
- **Analyze** the program
- **Evaluate** the correctness of the program and the accuracy of the algorithm.

4(b)

Hopfield Network Algorithm

---

1. Assign connection weights

$$w_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s & i \neq j \\ 0 & i = j, 0 \leq i, j \leq M-1 \end{cases}$$

where w_{ij} is the connection weight between node i and node j, and x_i^s is element i of the exemplar pattern for class s, and is either $+1$ or -1. There are M patterns, from 0 to $M-1$, in total. The thresholds of the units are zero.

2. Initialise with unknown pattern

$$\mu_i(0) = x_i \qquad 0 \leq i \leq N-1$$

where $\mu_i(t)$ is the output of node i at time t.

3. Iterate until convergence

$$\mu_i(t+1) = f_h\left[\sum_{i=0}^{N-1} w_{ij}\mu_j(t)\right] \qquad 0 \leq j \leq N-1$$

The function f_h is the hard-limiting non-linearity, the step function, as in figure 3.3. Repeat the iteration until the outputs from the nodes remain unchanged.

Your task is to:

- **Apply and Design** a program for your dataset
- **Analyze** the program
- **Evaluate** the correctness of the program and the accuracy of the algorithm.