Marking Scheme

Exam Paper
BSc CE

Neural Networks and Fuzzy Logic (630514)

Second Exam          First semester          Date: 27/12/2015
Section 1
Weighting 20% of the module total

Lecturer: Dr. Qadri Hamarsheh
Coordinator: Dr. Qadri Hamarsheh
Internal Examiner: Dr. Mohammed Mahdi
The presented exam questions are organized to overcome course material through 4 questions. The all questions are compulsory requested to be answered.

**Marking Assignments**

**Question 1** This question is attributed with 7 marks if answered properly; the answers are as following:

1) Which of the following equations is the best description of the *Perceptron Learning Rule*?
   - **a)** \( \Delta W_k = \eta y_k x \)
   - **b)** \( \Delta W_k = \eta (x - W_k) \)
   - **c)** \( \Delta W_k = \eta (d_k - y_k)x \)
   - **d)** \( \Delta W_j = \eta_j (x - W_j) \), where \( \eta_j < \eta \) and \( j \neq k \)

Where \( x \) is the input vector, \( \eta \) is the learning rate, \( W_k \) is the weight vector, \( d_k \) is the target output, and \( y_k \) is the actual output for unit \( k \).

2) In the backpropagation algorithm, how is the *error function* usually defined?
   - **a)** \( \frac{1}{2} \sum_j (\text{weight}_j \times \text{input}_j) \) for all inputs \( j \)
   - **b)** \( \frac{1}{2} \sum_j (\text{target}_j - \text{output}_j)^2 \) for all outputs \( j \)
   - **c)** \( \frac{1}{2} \sum_j (\text{target}_j - \text{output}_j) \) for all outputs \( j \)
   - **d)** *None of above*

3) A Hopfield network has 10 neurons. How many *adjustable parameters* does this network contain?
   - **a)** 45
   - **b)** 90
   - **c)** 100
   - **d)** 1024

4) Give the *equation* that can be used to convert the unipolar binary data \( (x) \) to bipolar binary data \( (y) \).
   - **a)** \( y = 2x \)
   - **b)** \( y = -2x - 1 \)
   - **c)** \( y = 2x + 1 \)
   - **d)** \( y = 2x - 1 \)

5) If the associated pattern pairs \( (x, y) \) are different and if the model recalls a \( y \) given an \( x \) or vice versa, then it is termed as
   - **a)** *Auto correlator*
   - **b)** Auto-associative memory
   - **c)** *Heteroassociative memory*
   - **d)** Double associative memory

6) An advantage with *gradient descent* based methods, such as back propagation, is that they cannot get stuck in local minima.
   - **a)** True
   - **b)** False

7) The second stage of back propagation training is ____________
   - **a)** initialization weights
   - **b)** back propagation of errors
   - **c)** feed forward
   - **d)** updating of weights and bias
Question 2 This question is attributed with 3 marks if answered properly; the answers are as following:

**Solution**

Calculate the weight matrix from the reference vectors, using the following rule.

\[ W = p_1(p_1)^T + p_2(p_2)^T - MI = \]

\[
\begin{bmatrix}
1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 \\
-1 & -1 & 1 & 1 \\
-1 & -1 & 1 & 1 \\
\end{bmatrix}
+ 
\begin{bmatrix}
1 & -1 & 1 & -1 \\
-1 & 1 & -1 & 1 \\
1 & -1 & 1 & -1 \\
-1 & 1 & -1 & 1 \\
\end{bmatrix} 
- 2 \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
= \]

\[
\begin{bmatrix}
2 & 0 & 0 & -2 \\
0 & 2 & -2 & 0 \\
0 & -2 & 2 & 0 \\
-2 & 0 & 0 & 2 \\
\end{bmatrix}
- \begin{bmatrix}
2 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 0 & 2 & 0 \\
0 & 0 & 0 & 2 \\
\end{bmatrix}
\]

Question 3 This question is attributed with 4 marks if answered properly; the answers are as following:

**Solution**

Bidirectional associative memory training algorithm:

1) **Storage (learning):** In the learning step for BAM we need to find weight matrix between, \( M \) pairs of patterns (fundamental memories) are stored in the synaptic weights of the network according to the equation:

\[ W = \sum_{m=1}^{M} X_m Y_m^T \]

2) **Testing**
   - We need to confirm that the BAM is able to
     - recall \( Y_m \) when presented \( X_m \).
     - recall \( X_m \) when presented \( Y_m \).

   Using
   \[ Y_m = \text{sign}(W^T X_m), \quad m = 1.2,...,M \]
   And using
   \[ X_m = \text{sign}(W Y_m), \quad m = 1.2,...,M \]

3) **Retrieval**
   - Present an unknown vector (probe) \( X \) (corrupted or incomplete version of a pattern from set A or B) to the BAM and retrieve a stored association:

   \[ X \neq X_m, \quad m = 1,2,...,M \]

   - **Initialize the BAM:**
     \[ X(0) = X, \quad p = 0 \]

   - **Calculate the BAM output at iteration \( p \):**
     \[ Y(p) = \text{sign}[W^T X(p)] \]

   - **Update the input vector \( X(p) \):**
     \[ X(p + 1) = \text{sign}[W Y(p)] \]

   - Repeat the iteration until convergence, when input and output remain unchanged.

Question 4 This question is attributed with 6 marks if answered properly; the answers are as following:

**Solution**

a) **Step 1: Begin training.**
   Step 2: For the first vector, \((1, 1, 0, 0)\), do Steps 3-5.

Step 3:

\[ D(1) = (.2 - 1)^2 + (.6 - 1)^2 + (.5 - 0)^2 + (.9 - 0)^2 = 1.86; \]
\[ D(2) = (.8 - 1)^2 + (.4 - 1)^2 + (.7 - 0)^2 + (.3 - 0)^2 = 0.98. \]

Step 4: The input vector is closest to output node 2, so
Step 5: The weights on the winning unit are updated:

\[
\begin{align*}
    w_{i2}^{\text{new}} &= w_{i2}^{\text{old}} + 0.6 \left[ x_i - w_{i2}^{\text{old}} \right] \\
    &= 0.4 w_{i2}^{\text{old}} + 0.6 x_i.
\end{align*}
\]

This gives the weight matrix

\[
\begin{bmatrix}
    0.2 & 0.92 \\
    0.6 & 0.76 \\
    0.5 & 0.28 \\
    0.9 & 0.12
\end{bmatrix}
\]

b) %Load the data
\[
data1 = [1, 1, 0; 0, 0, 1; 1, 0, 0; 0, 0, 1]';
\]
% Create a Self-Organizing Map
\[
dimension1 = 1;
dimension2 = 2;
net = selforgmap ([dimension1 dimension2]);
\]
% Train the Network
\[
[net, tr] = train (net, data1);
y = net (data1);
\]
% View the Network
\[
view (net)
\]
% Plot results using different SOM plots
\[
figure, plotsomtop (net)
figure, plotsomnc (net)
figure, plotsomnd (net)
figure, plotsomplanes (net)
figure, plotsomhits (net, data1)
figure, plotsompos (net, data1)
\]