**COMP 3710 Applied Artificial Intelligence**

**Seminar/Lab 8.**

**Backpropagation Neural Network**

1. Objectives

* Image recognition with a backpropagation neural network

1. We are planning to train a backpropagation neural network to recognize two symbols, ‘|’ and ‘\_’. The network has
   * Input layer – 4 nodes
   * Hidden layer – 2 nodes
   * Output layer – 2 nodes
   * Initial random weights – [0, 1]
   * Input threshold – 0.1
   * Learning rate – 0.2
   * Error = correct value – output
   * The output of 0 and 1 is considered as ‘|’, and the output of 1 and 0 is considered ‘\_’.
   * Activation function: Step function with 0.5 just for this exercise.

Y(X) = 1 if X >= 0.7;

Y(X) = 0.5 if X >= 0;

Y(X) = 0 if X < 0;

Here is the algorithm.

For each training image, Image

Output\_I = Input = Image;

For each node, j, in the hidden layer

X[j] = sum of Weight\_IH[i][j] \* Output\_I[i];

X[j] -= Threshold[j];

Output\_H[j] = activate(X[j]);

For each node, j, in the output layer

X[j] = sum of Weight\_HO[i][j] \* Output\_H[i];

X[j] -= Threshold[j];

Output\_O[j] = activate(X[j]);

Output = Output\_O;

For each node, j, in the output layer

Error[j] = Expected[j] – Output[j];

For each node, j, in the output layer

Delta\_O[j] = Output\_O[j] \* (1 – Output\_O[j]) \* Error[j];

For each node, j, in the hidden layer

D[j] = sum of Weight\_HO[j][k] \* Delta\_O[k];

Delta\_H[j] = Output\_H[j] \* (1 – Output\_H[j]) \* D[j];

For each node, j, in the hidden layer

For each node, k, in the output layer

Weight\_HO[j][k] += Alpha \* Output\_H[j] \* Delta\_O[k];

For each node, j, in the input layer

For each node, k, in the hidden layer

Weight\_IH[j][k] += Alpha \* Output\_I[j] \* Delta\_H[k];

Here is a training data set.

|  |  |
| --- | --- |
| Pattern | Class |
| [0, 1, 0, 1] | [0, 1] |
| [1, 0, 1, 0] | [0, 1] |
| [1, 1, 0, 0] | [1, 0] |
| [0, 0, 1, 1] | [1, 0] |

Here are initial weights.

|  |
| --- |
| W\_IH[4×2] |
| [[0.3, 0.6],  [0.2, 0.1],  [0.5, 0.4],  [0.1, 0.9]] |

|  |
| --- |
| W\_HO[2×2] |
| [[0.3, 0.6],  [0.1, 0.9]] |

Let’s simulate the training of the above BNN. Just one epoch.

For Input[0]: [0, 1, 0, 1]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Input | O\_I | Weighted  \_sum\_Hidden | O\_H | Weighted  \_sum\_Output | O\_O | E |
| [0,  1,  0,  1] | [0,  1,  0,  1] | X = 0.3 \* 0 + 0.2 \* 1 + 0.5 \* 0 + 0.1 \* 1 = 0.3 | X = X – 0.1 = 0.2 | X = 0.3 \* 0.5 + 0.1 \* 1 = 0.25 | X = X – 0.1 = 0.15 | E[0] = 0 – 0.5 = –0.5 |
| O\_H[0] = Y(0.2) = 0.5 | O\_O[0] = Y(X) = Y(0.15) = 0.5 |
|  |  |  |  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Delta\_Output | Delta\_Hidden | Weight\_HOL | Weights\_IHL |
| D\_O[0] = O\_O[0] \* (1-O\_O[0]) \* E[0] = 0.5 \* (1 – 0.5) \* -0.5 = -0.125 | D\_H[0] = O\_H[0] \* (1 - O\_H[0]) \* (W\_HO[0][0] \* D\_O[0] + W\_HO[0][1] \* D\_O[1] ) = 0.5 \* 0.5 \* (0.3 \* -0.125 + 0.6 \* 0) = -0.009375 | W\_HO[0][0] = W\_HO[0][0] + Alpha \* O\_H[0] \* D\_O[0] = 0.3 + 0.2 \* 0.5 \* -0.125 = 0.275 | W\_IH[0][0] = 0.3 + 0.2 \* 0 \*  -0.009375 = 0.3 |
| W\_IH[0][1] = 0.6 + 0.2 \* 0 \* 0 = 0.6 |
| W\_HO[0][1] = 0.6 + 0.2 \* 0.5 \* 0 = 0.6 | W\_IH[1][0] = 0.2 + 0.2 \* 1 \*  -0.009375 = 0.198125 |
| W\_IH[1][1] = 0.1 + 0.2 \* 1 \* 0 = 0 |
|  |  |  |  |
|  |
|  |  |
|  |

Here is what you need to do.

* Complete the above simulation
* Continue the above simulation with the second image, [1, 0, 1, 0].

1. Implementation of BNN

Sigmoid activation function, not step function, should be used in this NN. See Slide 24 and 25.

You need complete the following functions.

* train\_NN()
* nn\_process\_input()
* nn\_adjust\_weights()

1. Submission
   * The title of the mail should include your name, id, and COMP 3710.

* You need to submit a document file for 2), not image of hand-written document.
  + Total marks: 5
  + Due: 6:00 PM, November 6, 2017
* You need to submit the application in 4) by email, with the entire screen shot, not just window content, that shows how your application works. (Any submission with a window content shot will not be accepted.)
  + **Please do not share the code.** Any violation could cause a very serious problem.
  + Total marks: 15
    - * Any syntax error: 0 mark
      * No completion: 0 mark
        + Generally no partial marks for any code that cannot find a solution. You really need to complete this programming.
  + Due: 6:00 PM, November 8, 2017
* Any late submission will NOT be accepted.