ECE 4850/7650 Applied Computational Intelligence

Project 3

Backpropagation Analysis

Fall 2017

Due Date: 10:00 am, Monday 6 November 2017

1. For both ECE 4850 and ECE 7650: Implement a Backpropagation ANN to solve the NXOR problem: Use the following data set:

x_2	x_1	y
0	0	1
0	1	0
1	0	0
1	1	1

   1. Create a (3,4,1) architecture: use two input neurons and one bias, three hidden neurons and one bias, and one output neuron, i.e., (3,4,1).
   2. (**1 mark**) Derive the forward and backward propagation equations for this architecture in a manner similar to that done in the lecture slides for the (3,3,1) architecture (Slide 19).
   3. (**2 marks**) Test your implementation for correctness by computing the weights using pencil, paper, and calculator calculations for the 1st iteration of the algorithm (show your work), and compare your hand calculations with the weights obtained after the 1st iteration of your code. Use the training example {1 0 0}. Randomize your weights between the range {-1 to +1}. Submit your hand calculations and the Matlab code for the 1st iteration, including the value of each randomized weight, and the value of each weight after the 1st iteration.
   4. (**3 marks**) After a sufficient number of epochs are run to obtain 100% correct classification, examine and analyze the activations of the hidden layer, i.e., $a_1^{(2)}$, $a_2^{(2)}$, and $a_3^{(2)}$ to determine the nature of the extracted features. Give a truth table similar to that shown on Slide 19 of the lecture notes to show the nature of the extracted features. Are the extracted features the same as that given on Slide 25? Discuss. Can you suggest a way to run the algorithm so that it will produce the same features as those given on Slide 25? If so test and verify it. Submit your Matlab code, the final value of the weights, and a description of the extracted features to the TA for marking.

2. For ECE 7650 students only: Give the equation for $\delta_1^{(3)} = \frac{\partial J}{\partial a_1^{(3)}}$, where $J(\theta)$ is the logistic cost function: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} -y^i \log\left(h_\theta\left(x^i\right)\right) - (1-y^i)\log\left(1-h_\theta\left(x^i\right)\right)$. Run your Matlab code using the logistic cost function and compare the results with that obtained using the Euclidean cost function as done in (1) above. Submit your solutions to me for marking.