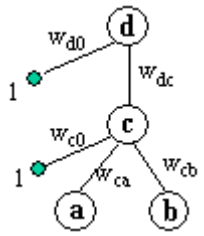
CSC 578 Quiz #2

Create a/one .**docx** or .**pdf** file containing all of your answers. The file must start with **your name**, **the course and section number in which you are registered** and the **assignment name/number** (e.g. "CSC 578 Quiz #2"). Files without those information will be returned ungraded.

Do all questions.

1. Mitchell's book, **Exercise 4.5**.
   Comments:
   - The question says a 'gradient descent training rule' -- it is implying the Gradient Descent algorithm of Table 4.1 (p. 93). Basically you are supposed to give a new rule/algorithm when the neuron's activation function is changed from linear (as in Table 4.1) to the one given in the question.
   - However the only difference in the rule/algorithm would be the weight update rule, that is (T4.1). So you can only give the **new weight update formula, for delta_w_i**.
   - But note the new formula should have **two cases**, because the w_0 term is different from all other w_1,..., wn terms.
   - Before giving the new rule, SHOW YOUR DERIVATION of the new weight update formula.

2. Mitchell's book, **Exercise 4.7**.



With the use of "threshold weight for unit x", you can assume the network as above, where the nodes in green are threshold units and their values are 1.

Comments:

- Translate "an iteration" to be an execution of the "For each <x,t> in training_examples" loop in Table 4.2 (p. 98) -- for one pattern (NOT a whole training set). So "the first two training iterations" in the question means you apply the algorithm for each of the two patterns ONCE (i.e., one time through the training set).
- Also assume the weight update for the 0th iteration (before the first pattern) is 0 for all weights. That means for the first iteration/pattern, the value of the momentum term is 0, while for the second iteration/pattern, the momentum terms uses the weight update from the first iteration/pattern.

3. Mitchell's book, **Exercise 4.8**.
   Comments:
   - This question is similar to Exercise 4.5 in that it is asking you to change the neuron's activation function. But for this question, the activation function is tanh, and the algorithm is Backprop (BP), shown in Table 4.2 (p. 98).
   - Also the BP algorithm is formulated differently from the algorithm for linear unit, so you must give **a whole new description of the algorithm**, in particular the **four steps in the loop** after "For each <x, t> in training_examples, Do".
   - Before giving the new algorithm/steps, SHOW YOUR DERIVATION of them.