CSC 578 Quiz#2 Sample Solutions

1. Mitchell's book 4.5

The gradient descent training rule for a single unit with output o, where

$$o = w_0 + w_1 x_1 + w_1 x_1^2 + w_2 x_2 + w_2 x_2^2 + \ldots + w_n x_n + w_n x_n^2$$

First derive the gradient of the error function E.

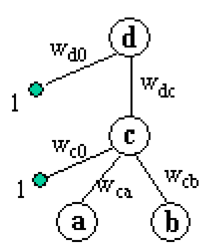
$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \cdot \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d \in D} 2 \cdot (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \\
&= \sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - (w_0 + w_1 x_{1,d} + w_1 x_{1,d}^2 + \ldots + w_n x_{n,d} + w_n x_{n,d}^2)) \\
&= \begin{cases} \sum_{d \in D} (t_d - o_d)(-1) \cdots \text{for } w_0 \\ \sum_{d \in D} (t_d - o_d)(-x_{i,d} - x_{i,d}^2) \cdots \text{for } w_i, 1 \le i \le n \end{cases}
\end{aligned}$$

Then the weight update rule (the delta portion) is the negative of the gradient (i.e., descent) multiplied by the learning rate.

$$\Delta w_i = \begin{cases} \eta \cdot \sum_{d \in D} (t_d - o_d) \cdots \text{for } w_0 \\ \eta \cdot \sum_{d \in D} (t_d - o_d)(x_{i,d} + x_{i,d}^2) \cdots \text{for } w_i, 1 \le i \le n \end{cases}$$

2. Mitchell's book 4.7

The network is



- First pattern <<1, 0>, 1>

$$o_c = \frac{1}{1 + EXP\{-[(0.1) + (0.1)(1) + (0.1)(0)]\}} = \frac{1}{1 + EXP\{-0.2\}} = 0.5498$$

$$o_d = \frac{1}{1 + EXP\{-[(0.1) + (0.1)(0.5498]\}} = \frac{1}{1 + EXP\{-0.15498\}} = 0.5387$$

$$\delta_d = (0.5387)(1 - 0.5387)(1 - 0.5387) = (0.5387)(0.4613)(0.4613) = 0.1146$$

$$\delta_c = (0.5498)(1 - 0.5498)(0.1)(0.1146) = (0.5498)(0.4502)(0.1)(0.1146) = 0.002837$$

Updated weights are:

$$\Delta w_{d0} = (0.3)(0.1146)(1) = 0.03438 \quad therefore \quad w_{d0} = 0.1 + 0.03438 = 0.13438$$

$$\Delta w_{dc} = (0.3)(0.1146)(0.5498) = 0.0189 \quad therefore \quad w_{dc} = 0.1 + 0.0189 = 0.1189$$

$$\Delta w_{c0} = (0.3)(0.002837)(1) = 0.0008511 \quad therefore \quad w_{c0} = 0.1 + 0.0008511 = 0.1008511$$

$$\Delta w_{ca} = (0.3)(0.002837)(1) = 0.0008511 \quad therefore \quad w_{ca} = 0.1 + 0.0008511 = 0.1008511$$

$$\Delta w_{cb} = (0.3)(0.002837)(0) = 0.0 \quad therefore \quad w_{cb} = 0.1 + 0.0 = 0.1$$

- Second pattern <<0, 1>, 0>

$$o_c = \frac{1}{1 + EXP\{-[(0.1008511) + (0.1008511)(0) + (0.1)(1)]\}} = \frac{1}{1 + EXP\{-0.2008511\}} = 0.5500$$

$$o_d = \frac{1}{1 + EXP\{-[(0.13438) + (0.1189)(0.5500)]\}} = \frac{1}{1 + EXP\{-0.1198\}} = 0.5498$$

$$\delta_d = (0.5498)(1 - 0.5498)(0 - 0.5498) = (0.5498)(0.4502)(-0.5498) = -0.1361$$

$$\delta_c = (0.5500)(1 - 0.5500)(0.1189)(-0.1361) = (0.5500)(0.4500)(0.1189)(-0.1361) = -0.0040$$

Updated weights are:

$$\Delta w_{d0} = (0.3)(-0.1361)(1) + (0.9)(0.03438) = -0.0099 \quad \text{So,} \quad w_{d0} = 0.13438 - 0.0099 = 0.12448$$

$$\Delta w_{dc} = (0.3)(-0.1361)(0.5500) + (0.9)(0.0189) = -0.0055 \quad \text{So,} \quad w_{dc} = 0.1189 - 0.0055 = 0.1135$$

$$\Delta w_{c0} = (0.3)(-0.0040)(1) + (0.9)(0.0008511) = -0.00043 \quad \text{So,} \quad w_{c0} = 0.1008511 - 0.00043 = 0.10042$$

$$\Delta w_{ca} = (0.3)(-0.0040)(0) + (0.9)(0.0008511) = 0.000766 \quad \text{So,} \quad w_{ca} = 0.1008511 + 0.000766 = 0.1016$$

$$\Delta w_{cb} = (0.3)(-0.0040)(1) + (0.9)(0) = -0.0012 \quad \text{So,} \quad w_{cb} = 0.1 - 0.0012 = 0.0988$$

3. Mitchell's book 4.8

In a network in which the function tanh is used in place of sigmoid, the output/activation function of a node (hidden and output) is, as given in the question, $o = \tanh(\vec{w} \cdot \vec{x})$.
Also as given in the question, the first derivative of the tanh function (tanh') is
$\tanh'(y) = 1 - \tanh^2(y)$.
Note here that I'm using y instead of x in this formula (as given in the question) in order to avoid confusion between the input vector ($\vec{x}$) and the input variable in tanh.

First derive the gradient of the error function E for a tanh unit

$$
\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \cdot \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d \in D} 2 \cdot (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \\
&= \sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i} (-o_d) \\
&= -\sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i} (\tanh(\vec{w} \cdot \vec{x}_d)) \\
&= -\sum_{d \in D} (t_d - o_d) \cdot \tanh'(\vec{w} \cdot \vec{x}_d) \cdot \frac{\partial}{\partial w_i} (\vec{w} \cdot \vec{x}_d) \\
&= -\sum_{d \in D} (t_d - o_d) \cdot \tanh'(\vec{w} \cdot \vec{x}_d) \cdot (x_{i,d}) \\
&= -\sum_{d \in D} (t_d - o_d) \cdot (1 - \tanh^2(\vec{w} \cdot \vec{x}_d)) \cdot (x_{i,d}) \\
&= -\sum_{d \in D} (t_d - o_d) \cdot (1 - o_d^2) \cdot (x_{i,d})
\end{aligned}
$$

So the weight update rule becomes:

$\Delta w_i = \eta \cdot \sum_{d \in D} (t_d - o_d)(1 - o_d^2)(x_{i,d})$ ... for the Batch version

$\Delta w_i = \eta \cdot (t_d - o_d)(1 - o_d^2)(x_{i,d})$ ... for the Stochastic version

Therefore, the backpropagation algorithm (for the Stochastic version) becomes:

Initialize all weights to small random numbers.
Until satisfied, do

- For each training example $< \vec{x}, \vec{t} >$, do
    1. Input the training example to the network and compute the network outputs.
    2. For each output unit k,
       $\delta_k \leftarrow (t_k - o_k)(1 - o_k^2)$
    3. For each hidden unit h,
       $\delta_h \leftarrow (1 - o_h^2) \sum_{k \in Outputs} w_{k,h} \delta_k$
    4. Update each network weight $w_{i,j}$,
       $w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j}$ where
       $\Delta w_{i,j} \leftarrow \eta \cdot \delta_i \cdot x_j$