Let

$$T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots (x^{(m)}, y^{(m)})\} \tag{1}$$

be the training set, where $x^{(i)} \in \mathbb{R}^n$ is the *feature vector* and $y^{(i)}$ is the *target value* for $i-$th sample.
The **backpropagation algorithm** for computing the derivate updates in the *gradient descent* process is given by following steps -

1. Let $l \in \{1, 2, \dots, L\}$, where L is the total number of layers in the NN.
2. Set $\Delta_{ij}^{(l)} = 0$ for all i, j, l.
   This will contain updates for each weight. $\Delta_{ij}^{(l)}$ corresponds to the weight of unit i in layer l for the contribution of unit j in the previous layer.
3. for $i \in \{1, 2, \dots, m\}$
   - Set $a^{(1)} = x^{(i)}$
   - Perform forward propagation to compute activation of each layer $a^{(l)}$ for $l = 2, 3, \dots, L$, as

     $$a^{(l+1)} = g(a^{(l)}(\Theta^{(l)})^T)$$

     where g is the *activation function.*
   - Using $y^{(i)}$, compute the error of the last layer as

     $$\delta^{(L)} = a^{(L)} - y^{(i)}$$

   - Compute $\delta^{(l)}$ for $l = L-1, L-2, ..., 2$ as

     $$\delta^{(l)} = (\Theta^{(l)})^T \delta^{(l+1)} * (a^{(l)}(1 - a^{(l)}))$$

     where $*$ is the element-wise multiplication.
   - $\Delta_{ij}^{(l)} = \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$
4. If $j \neq 0$

   $$D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)} + \lambda \Theta_{ij}^{(l)}.$$

   else, if $j = 0$,

   $$D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)}$$

It can be checked that,

$$\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) = D_{ij}^{(l)}$$