Part 2. Multi Layer Networks
Sigmoid-Function for Continuous Output

\[ a = \sum_{i=0}^{n} w_i x_i \]

\[ O = \frac{1}{1 + e^{-a}} \]

Output between 0 and 1 (when \( a = \) negative infinity, \( O = 0 \); when \( a = \) positive infinity, \( O = 1 \).
Gradient Descent Learning Rule

- For each training example $X$, 
  - Let $O$ be the output (between 0 and 1) 
  - Let $T$ be the correct target value 
- Continuous output $O$ 
  - $a = w_1 x_1 + ... + w_n x_n + \theta$ 
  - $O = 1/(1+e^{-a})$ 
- Train the $w_i$’s such that they minimize the squared error 
  - $E[w_1,...,w_n] = \frac{1}{2} \sum_{k \in D} (T_k - O_k)^2$
  where $D$ is the set of training examples
Explanation: Gradient Descent Learning Rule

\[ \Delta w_i = \alpha O_k (1 - O_k) (T_k - O_k) x_i^k \]

- \( O_k \): Activation of pre-synaptic neuron
- \( w_i \): Weight
- \( x_i \): Input
- \( \alpha \): Learning rate
- \( \delta_k \): Error of post-synaptic neuron
- \( T_k \): Target output
- \( O_k \): Output of post-synaptic neuron
- \( x_i^k \): Activation of pre-synaptic neuron

The diagram illustrates the components and their relationships in the gradient descent learning rule.
Backpropagation Algorithm (Han, Figure 9.5)

- Initialize each $w_i$ to some small random value
- Until the termination condition is met, Do
  - For each training example $<(x_1,\ldots,x_n),t>$ Do
    - Input the instance $(x_1,\ldots,x_n)$ to the network and compute the network outputs $O_k$
    - For each output unit $k$
      - $Err_k = O_k(1-O_k)(t_k-O_k)$
    - For each hidden unit $h$
      - $Err_h = O_h(1-O_h) \sum_k w_{h,k} Err_k$
    - For each network weight $w_{i,j}$ Do
      - $w_{i,j} = w_{i,j} + \Delta w_{i,j}$ where
        $\Delta w_{i,j} = \alpha \text{Err}_j \ast O_i$
      - $\theta_j = \theta_j + \Delta \theta_j$ where
        $\Delta \theta_j = \alpha \text{Err}_j$
  - $\alpha$: is learning rate, set by the user;
Given the following neural network with initialized weights as in the picture (next page), we are trying to distinguish between nails and screws and an example of training tuples is as follows:

- T1{0.6, 0.1, nail}
- T2{0.2, 0.3, screw}

Let the learning rate (l) be 0.1. Do the forward propagation of the signals in the network using T1 as input, then perform the back propagation of the error. Show the changes of the weights. Given the new updated weights with T1, use T2 as input, show whether the predication is correct or not.
Multilayer Neural Network
Multilayer Neural Network

- **Answer:**

- First, use T1 as input and then perform the back propagation.
  - At Unit 3:
    - \( a_3 = x_1 w_{13} + x_2 w_{23} + \theta_3 = 0.14 \)
    - \( o_3 = \frac{1}{1 + e^{-a}} = 0.535 \)

- Similarly, at Unit 4,5,6:
  - \( a_4 = 0.22, \quad o_4 = 0.555 \)
  - \( a_5 = 0.64, \quad o_5 = 0.655 \)
  - \( a_6 = 0.1345, \quad o_6 = 0.534 \)
Multilayer Neural Network

- Now go back, perform the back propagation, starts at Unit 6:
  - $\text{Err}_6 = o_6(1-o_6)(t-o_6) = 0.534 \times (1-0.534) \times (1-0.534) = 0.116$
  - $\Delta w_{36} = \langle l \rangle \text{Err}_6 O_3 = 0.1 \times 0.116 \times 0.535 = 0.0062$
  - $w_{36} = w_{36} + \Delta w_{36} = -0.394$
  - $\Delta w_{46} = \langle l \rangle \text{Err}_6 O_4 = 0.1 \times 0.116 \times 0.555 = 0.0064$
  - $w_{46} = w_{46} + \Delta w_{46} = 0.1064$
  - $\Delta w_{56} = \langle l \rangle \text{Err}_6 O_5 = 0.1 \times 0.116 \times 0.655 = 0.0076$
  - $w_{56} = w_{56} + \Delta w_{56} = 0.6076$
  - $\theta_6 = \theta_6 + \langle l \rangle \text{Err}_6 = -0.1 + 0.1 \times 0.116 = -0.0884$
Multilayer Neural Network

- **Continue back propagation:**
  - **Error at Unit 3:**
    \[ \text{Err}_3 = o_3 (1 - o_3) (w_{36} \ \text{Err}_6) = 0.535 \times (1-0.535) \times (-0.394\times0.116) = -0.0114 \]
    \[ w_{13} = w_{13} + \Delta w_{13} = w_{13} + (l) \ \text{Err}_3X_1 = 0.1 + 0.1*(-0.0114) \times 0.6 = 0.09932 \]
    \[ w_{23} = w_{23} + \Delta w_{23} = w_{23} + (l) \ \text{Err}_3X_2 = -0.2 + 0.1*(-0.0114) \times 0.1 = -0.2001154 \]
    \[ \theta_3 = \theta_3 + (l) \ \text{Err}_3 = 0.1 + 0.1 \times (-0.0114) = 0.09886 \]
  - **Error at Unit 4:**
    \[ \text{Err}_4 = o_4 (1 - o_4) (w_{46} \ \text{Err}_6) = 0.555 \times (1-0.555) \times (-0.1064\times0.116) = 0.003 \]
    \[ w_{14} = w_{14} + \Delta w_{14} = w_{14} + (l) \ \text{Err}_4X_1 = 0 + 0.1*(-0.003) \times 0.6 = 0.00018 \]
    \[ w_{24} = w_{24} + \Delta w_{24} = w_{24} + (l) \ \text{Err}_4X_2 = 0.2 + 0.1*(-0.003) \times 0.1 = 0.20003 \]
    \[ \theta_4 = \theta_4 + (l) \ \text{Err}_4 = 0.2 + 0.1 \times (0.003) = 0.2003 \]
  - **Error at Unit 5:**
    \[ \text{Err}_5 = o_5 (1 - o_5) (w_{56} \ \text{Err}_6) = 0.655 \times (1-0.655) \times (-0.6076\times0.116) = 0.016 \]
    \[ w_{15} = w_{15} + \Delta w_{15} = w_{15} + (l) \ \text{Err}_5X_1 = 0.3 + 0.1*0.016 \times 0.6 = 0.30096 \]
    \[ w_{25} = w_{25} + \Delta w_{25} = w_{25} + (l) \ \text{Err}_5X_2 = -0.4 + 0.1*0.016 \times 0.1 = -0.39984 \]
    \[ \theta_5 = \theta_5 + (l) \ \text{Err}_5 = 0.5 + 0.1 \times 0.016 = 0.5016 \]
After T1, the updated values are as follows:

<table>
<thead>
<tr>
<th>$w_{13}$</th>
<th>$w_{14}$</th>
<th>$w_{15}$</th>
<th>$w_{23}$</th>
<th>$w_{24}$</th>
<th>$w_{25}$</th>
<th>$w_{36}$</th>
<th>$w_{46}$</th>
<th>$w_{56}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.09932</td>
<td>0.00018</td>
<td>0.30096</td>
<td>-0.2001154</td>
<td>0.20003</td>
<td>-0.39984</td>
<td>-0.394</td>
<td>0.1064</td>
<td>0.6076</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\theta_3$</th>
<th>$\theta_4$</th>
<th>$\theta_5$</th>
<th>$\theta_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.09886</td>
<td>0.2003</td>
<td>0.5016</td>
<td>-0.0884</td>
</tr>
</tbody>
</table>

Now, with the updated values, use T2 as input:

- **At Unit 3:**
  - $a_3 = x_1w_{13} + x_2w_{23} + \theta_3 = 0.0586898$
  - $o_3 = \frac{1}{1 + e^{-a}} = 0.515$
Similarly,

- $a_4 = 0.260345$, $o_4 = 0.565$
- $a_5 = 0.441852$, $o_5 = 0.6087$

At Unit 6:

- $a_6 = x_3w_{36} + x_4w_{46} + x_5w_{56} + \theta_6 = 0.13865$
- $o_6 = \frac{1}{1 + e^{-a}} = 0.5348$

Since $O_6$ is closer to 1, so the prediction should be nail, different from given “screw”.

So this predication is NOT correct.