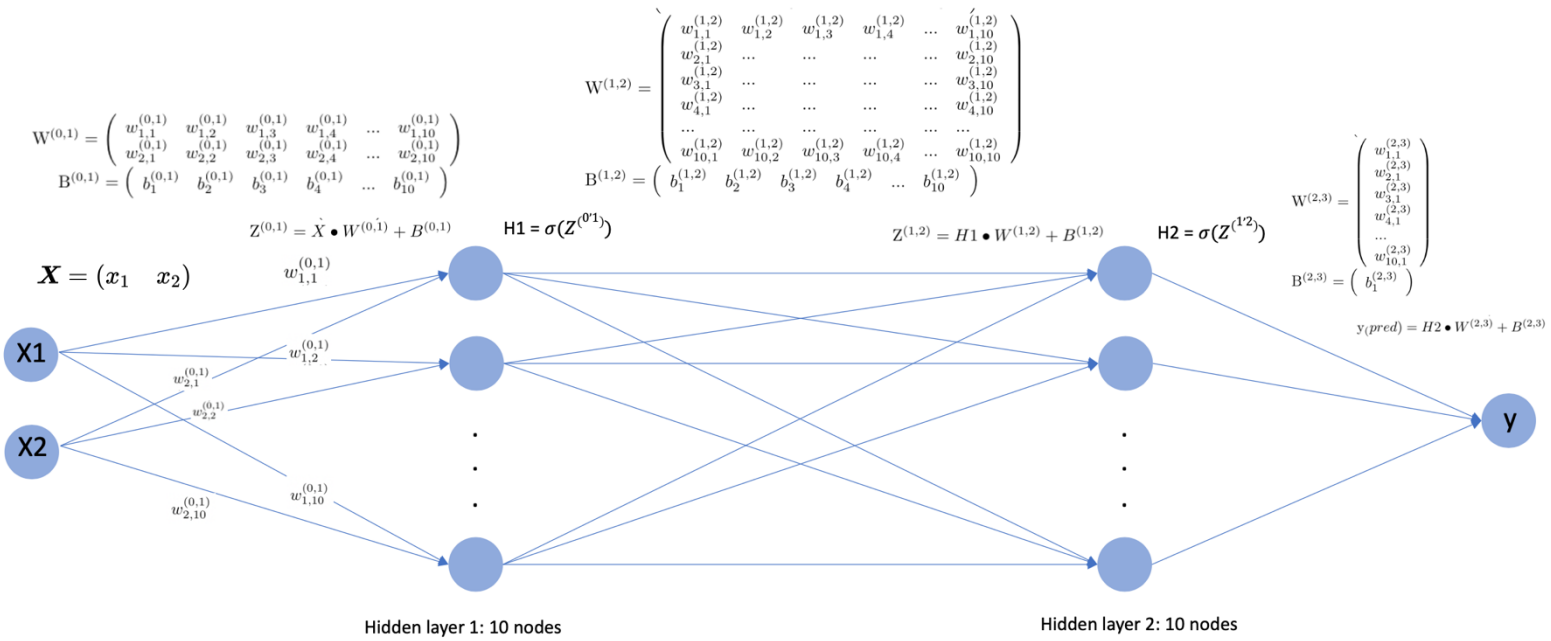
BS6207 **Assignment 1** **Jiao Yan**

A fully connected neural network has the architecture as such that all the nodes in one layer are **connected** to the neurons in the next layer. Do one-time forward propagation with one data point and compute the loss.

The forward propagation process is as follows:

$$Z^{(0,1)} = X \cdot W^{(0,1)} + B^{(0,1)} \qquad (1*10) = (1*2)\cdot(2*10) + (1*10)$$
$$H^{(1)} = \text{Sigmoid}\left(Z^{(0,1)}\right) \qquad (1*10)$$
$$Z^{(1,2)} = H1 \cdot W^{(1,2)} + B^{(1,2)} \qquad (1*10) = (1*10)\cdot(10*10) + (1*10)$$
$$H^{(2)} = \text{Sigmoid}\left(Z^{(1,2)}\right) \qquad (1*10)$$
$$y_{(}\text{pred}\,) = H2 \cdot W^{(2,3)} + B^{(2,3)} \qquad (1*1) = (1*10)\cdot(10*1) + (1*1)$$

The neural network is shown as below:



The loss function is defined as $L = (\hat{y} - y)^2$.

Then for the back propagation, compute the gradients using pytorch.autograd and my own implementation of the forward propagation and backpropagation algorithm from scratch.

Since we are trying to compare the gradients with 2 functions, the input X and true y and the initial weights and bias for 2 functions are the same.

Then the gradients of my implementation are computed as below:

$$\frac{\partial L}{\partial b_3} = \frac{\partial L}{\partial \hat{y}_3} \cdot \frac{\partial \hat{y}_3}{\partial b_3} = \frac{\partial L}{\partial \hat{y}_3} \times 1 = delta1$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial \hat{y}_3} \cdot \frac{\partial \hat{y}_3}{\partial w_3} = delta1 \cdot h_1$$

$$\frac{\partial L}{\partial b_2} = \left(\frac{\partial L}{\partial \hat{y}_3} \cdot \frac{\partial \hat{y}_3}{\partial h_2}\right) \cdot \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial b_2} = delta1 \cdot w_3 \times derivate\_sigmoid(h_2) \times 1 = delta2$$

$$\frac{\partial L}{\partial w_2} = \left(\frac{\partial L}{\partial \hat{y}_3} \cdot \frac{\partial \hat{y}_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2}\right) \cdot \frac{\partial z_2}{\partial w_2} = delta2 \cdot h_1$$

$$\frac{\partial \mathrm{L}}{\partial b_1} = \left(\frac{\partial \mathrm{L}}{\partial \hat{y}_3} \cdot \frac{\partial \hat{y}_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2}\right) \cdot \frac{\partial z_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial b_1} = delta2 \cdot w_2 \times derivate\_sigmoid(h_1) \times 1 = delta3$$

$$\frac{\partial \mathrm{L}}{\partial w_1} = \left(\frac{\partial \mathrm{L}}{\partial \hat{y}_3} \cdot \frac{\partial \hat{y}_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial z_1}\right) \cdot \frac{\partial z_1}{\partial w_1} = delta3 \cdot \mathrm{X}$$

The output gradients *torch_autograd.dat* and *my_autograd.dat* give the same values up to 7 decimal digits (since the dtype of the 2 functions are float32, which gives 6~7 digits of precision).