Training algorithm

① Initialize weights and learning rate (random small values are taken)

② Calculate net input to each hidden unit z_j, $j = 1 \ldots p$

$$(Z_j)_{in} = Z_{0j} + \sum_{i=1}^{n} x_i V_{ij}$$

③ Calculate outputs from hidden layer by applying activation functions over $(Z_{in})_j$, $Z_j = f((Z_{in})_j)$

④ These signals are sent as input signals for output unit

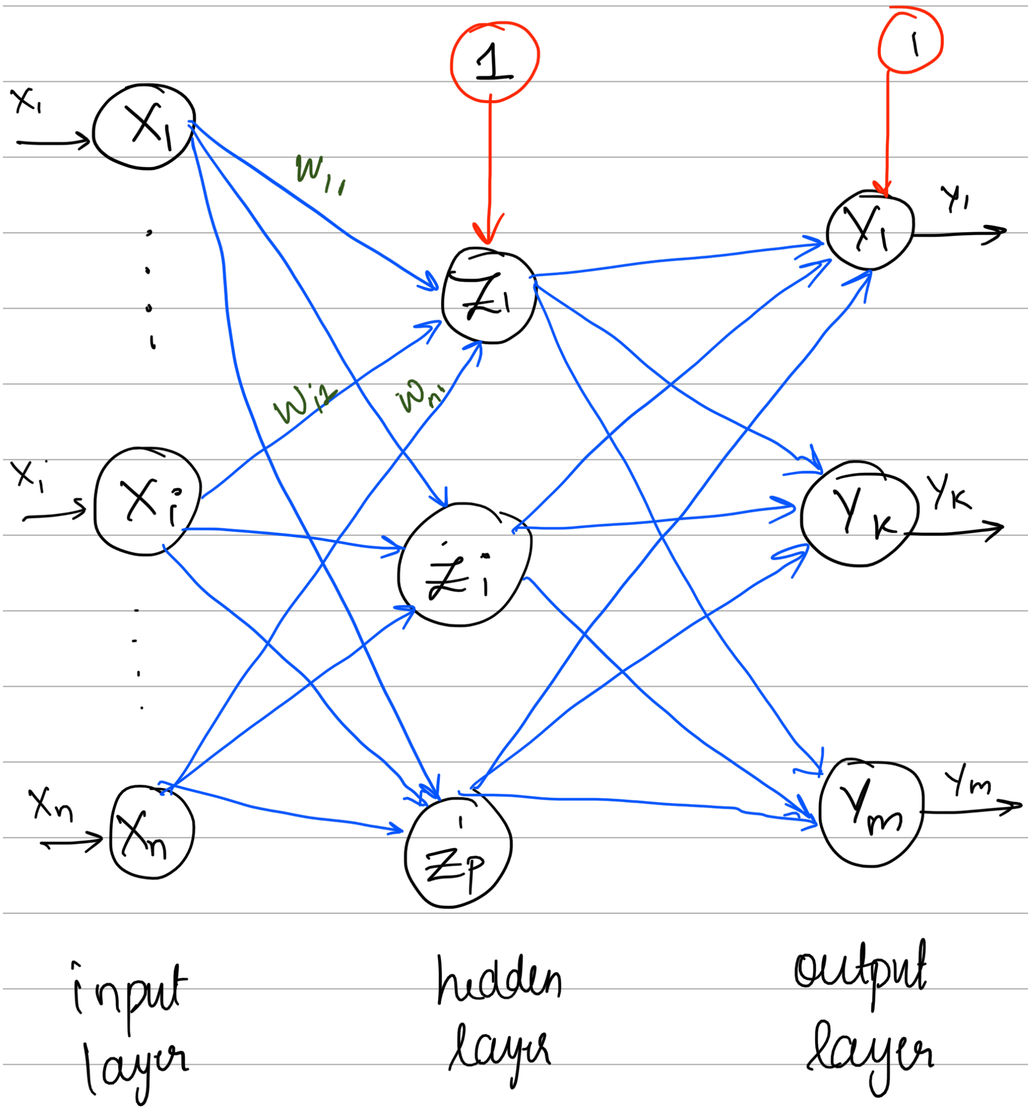
⑤ For each output unit y_k, $k = 1 \ldots m$, calculate net input

$$(y_{in})_k = W_{0k} + \sum_{j=1}^{p} Z_j \cdot W_{jk}$$

⑥ Apply activation function to compute output signal

$$y_k = f((y_{in})_k), \quad k = 1, 2 \ldots m$$

Architecture



(7) Each output unit y_k, $k = 1, 2 \ldots m$ receives a target pattern to the input training pattern and computes the error correction using:-

$$\delta_k = (t_k - y_k)\, f'(y_{in})_k$$

(8) On the basis of calculated error correction term, update the change in weights and bias :-

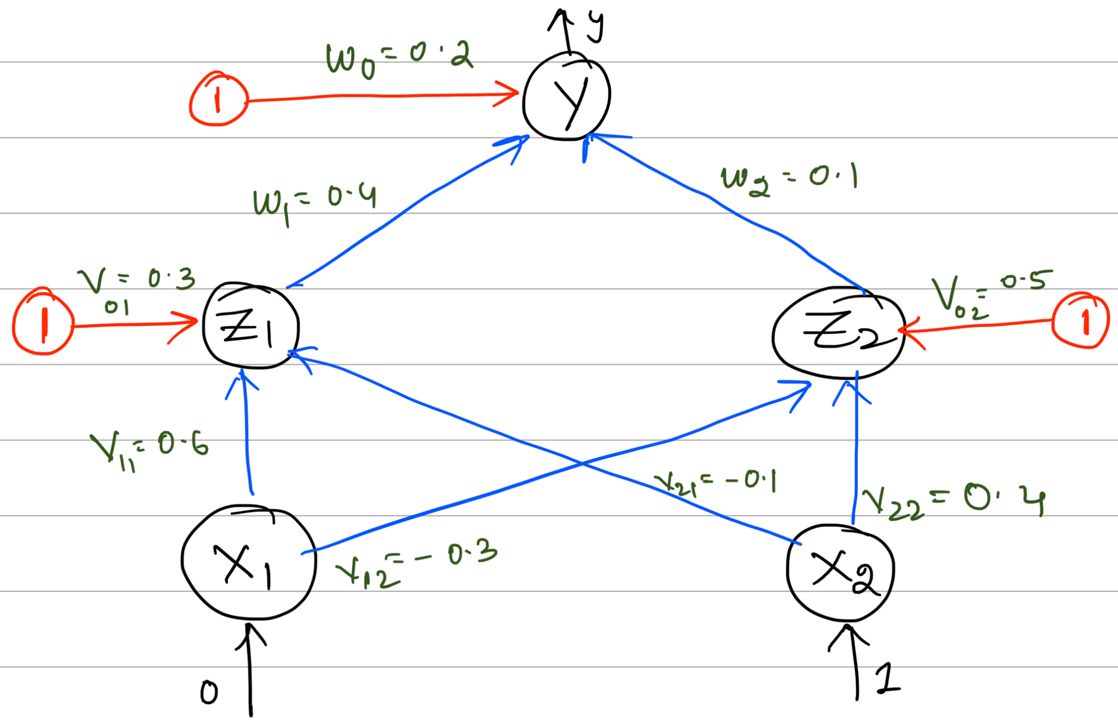
$$\Delta w_{jk} = \alpha \, \delta_j \, z_j$$

$$\Delta w_{0k} = \alpha \, \delta_k$$

(9) Send δ_k to the hidden layer backwards

(10) Update the weights and bias for output and hidden layer

Worked out Example



Target output = 1
learning rate $\alpha = 0.25$
Activation function : Sigmoid function $\frac{1}{1+e^{-x}}$

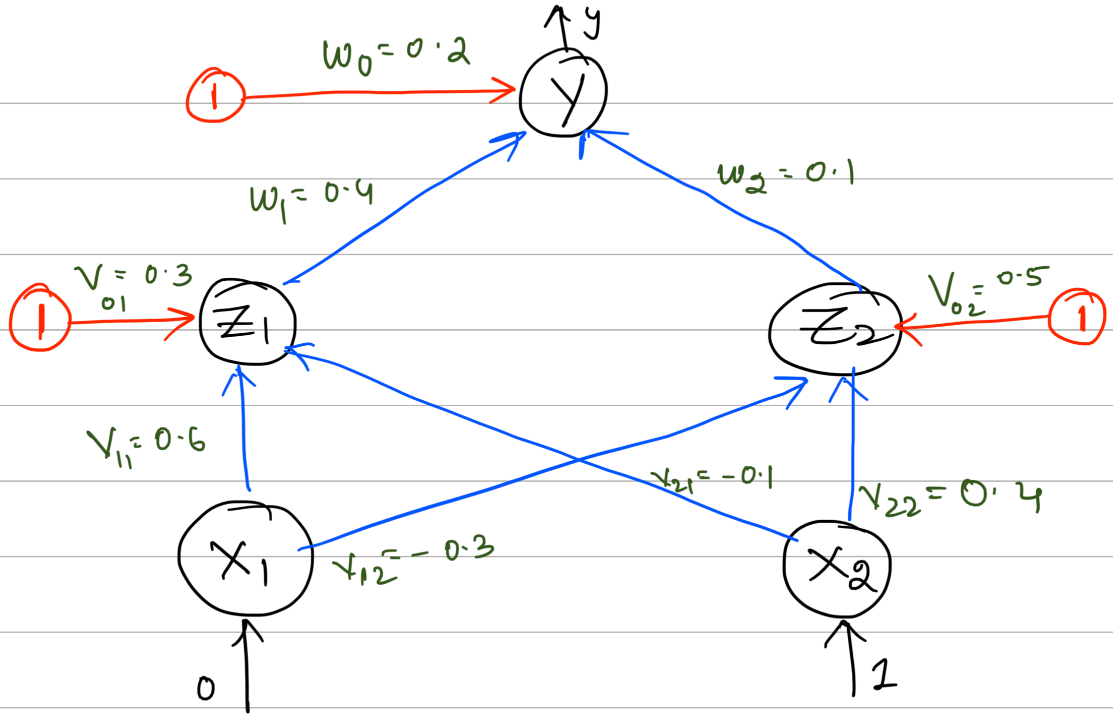
SOLUTION:

Step-1 Calculate net input for $Z1$:-

$$Z_{in1} = 1(V_{01}) + 0(V_{11}) + 1(V_{21})$$
$$Z_{in1} = 1(0.3) + 0(0.6) + 1(-0.1) = 0.2$$

Calculate net input for $Z2$:-

$$Z_{in2} = 1(V_{02}) + 0(V_{12}) + 1(V_{22})$$
$$= 1(0.5) + 0(-0.3) + 1(0.4) = 0.9$$



Target output = 1

learning rate $\alpha = 0.25$

Activation function : Sigmoid function $\frac{1}{1+e^{-x}}$

Sol^n continues:-

Next calculate output from hidden layer

Step-2

To do so apply Activation function to calculate output

$Z_{in_1} = 0.2$; $Z_{in_2} = 0.9$ (calculated in prev. step)

$$Z_1 = f(Z_{in_1}) = \frac{1}{1+e^{-Z_{in_1}}} = \frac{1}{1+e^{-0.2}} = 0.5498$$

$$Z_2 = f(Z_{in_2}) = \frac{1}{1+e^{-Z_{in_2}}} = \frac{1}{1+e^{-0.9}} = 0.7109$$

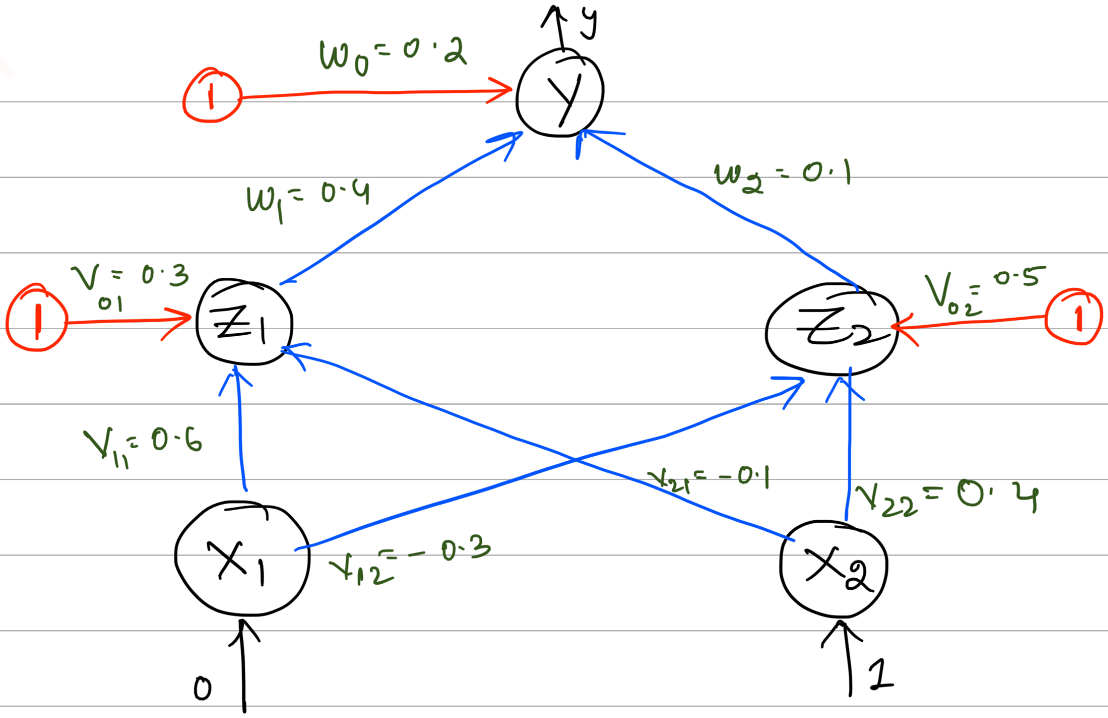
Step 3:

Calculate net input to output layer

$$y_{in} = 1(w_0) + Z_1 w_1 + Z_2 w_2.$$

$$= 1(0.2) + 0.5498 \times 0.4 + 0.7109 \times 0.1$$

$$= 0.09101$$



Target output = 1
learning rate $\alpha = 0.25$
Activation function : Sigmoid function $\frac{1}{1+e^{-x}}$

Step 4:

Calculate net output [for entire neural network]

$$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}} = \frac{1}{1+e^{-0.09101}} = 0.5227$$

$y_{in} = 0.09101$ [calculated in step 3]

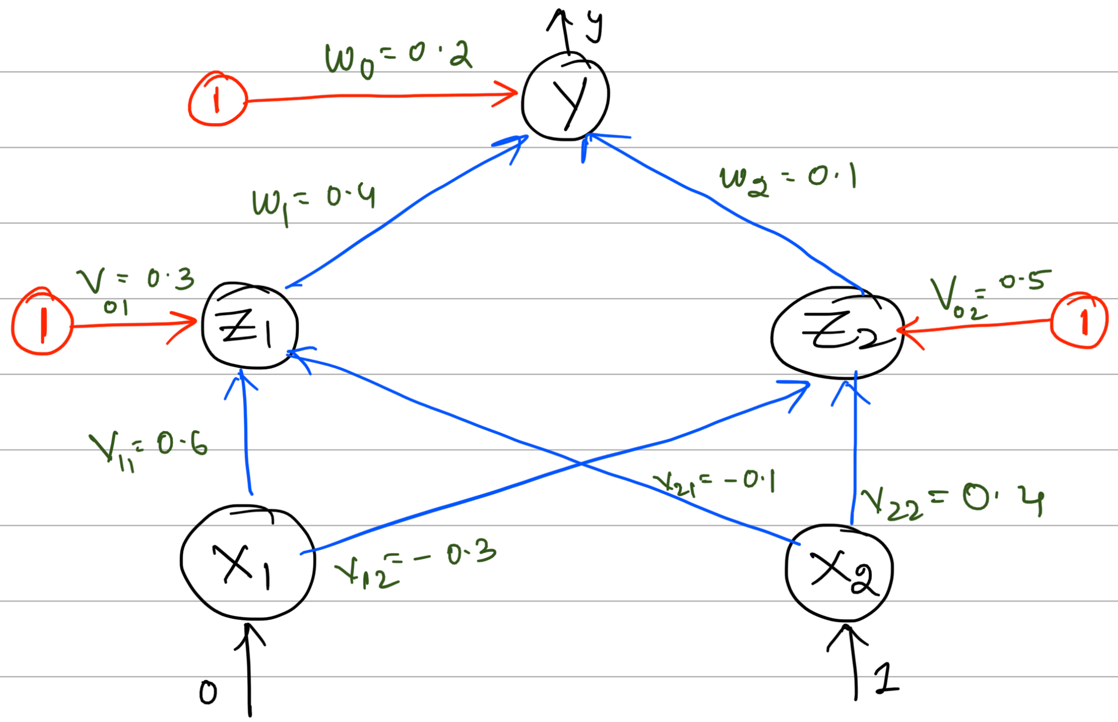
Target output = 1
Output obtained : 0.5227] error

Step 5

Calculate the δ at output layer

compute error portion δ_k:

$$\delta_k = (t_k - y_k)\, f'(y_{ink})$$



Target output = 1

learning rate $\alpha = 0.25$

Activation function : Sigmoid function $\frac{1}{1+e^{-x}}$

Steps :

$$\delta_k = (t_k - y_k) f'(y_{ink})$$

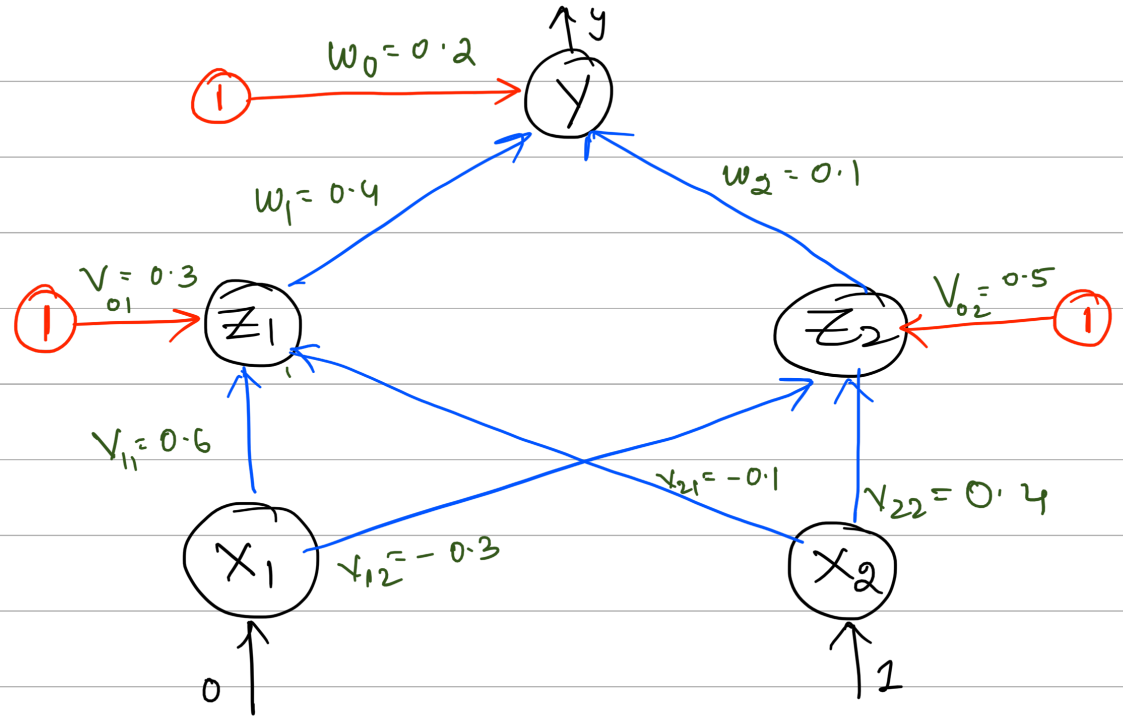
$$f'(y_{in}) = f(y_{in})[1 - f(y_{in})]$$

$$f'(y_{in}) = 0.5227[1 - 0.5227] = 0.2495$$

since $\sigma(x) = \frac{1}{1+e^{-x}}$ then $\sigma'(x) = \sigma(x)[1 - \sigma(x)]$

$$\delta_1 = (1 - 0.5227)(0.2495) = 0.1191$$

Step-6 Find the changes in weights between hidden and output layer –



Target output = 1

learning rate $\alpha = 0.25$

Activation function: Sigmoid function $\frac{1}{1+e^{-x}}$

Step 6:

$$\Delta W_1 = \alpha \delta_1 Z_1 = 0.25[0.1191][0.5498]$$
$$= 0.0164$$

$$\Delta W_2 = \alpha \delta_1 Z_2 = 0.25[0.1191][0.7109] = 0.02117$$

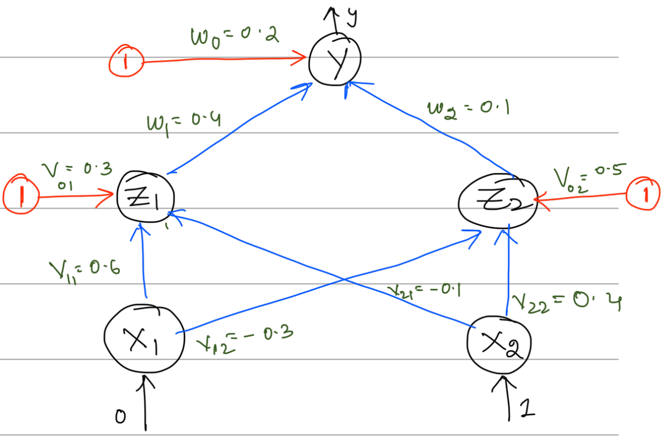
$$\Delta W_0 = \alpha \delta_1 (1) = 0.25[0.1191] = 0.02978$$

Step 7:

Compute error portion at hidden layer

$$\delta_j = \delta_{inj} (f'(z_{inj}))$$

Compute error portion δ_j between input and hidden layer (1 to 2)



$$\delta_j = \delta_{in_j} \left(f'(z_{in_j}) \right)$$

$$\delta_{in_j} = \sum_{k=1}^{m} \delta_k w_{jk}$$

$$\delta_{in1} = \delta_1 w_{11} = 0.1191 \times 0.4 = 0.04764$$

$$\delta_{in2} = \delta_1 w_{21} = 0.1191 \times 0.1 = 0.01191$$

error, $\delta_1 = \delta_{in1} f'(z_{in1})$

$$f'(z_{in1}) = f(z_{in1})(1 - f(z_{in1}))$$
$$= 0.5498(1 - 0.5498) = 0.2475$$

$$\delta_1 = \delta_{in1} f'(z_{in1})$$

$$= 0.04764(0.2475) = 0.0118$$

error, $\delta_2 = \delta_{in2} f'(z_{in2})$

$$f'(z_{in2}) = f(z_{in2})(1 - f(z_{in2}))$$
$$= 0.7109(1 - 7109) = 0.2055$$

$$\delta_2 = \delta_{in2}(0.2055)$$
$$= 0.1191 \times 0.2055 = 0.00245$$

Use δ_1, δ_2 and update weights between input and hidden layer

Find changes in weights between input and hidden layer:-

Step 8

$\Delta V_{11} = \alpha \delta_1 x_1 = 0.25 \times 0.0118 \times 0 = 0$

$\Delta V_{21} = \alpha \delta_1 x_2 = 0.25 \times 0.0118 \times 1 = 0.00295$

$\Delta V_{01} = \alpha \delta_1 = 0.25 \times 0.0118 = 0.00295$

$\Delta V_{12} = \alpha \delta_2 x_1 = 0.25 \times 0.00245 \times 0 = 0$

$\Delta V_{22} = \alpha \delta_2 x_2 = 0.25 \times 0.00245 \times 1 = 0.0006125$

$\Delta V_{02} = \alpha \delta_2 = 0.25 \times 0.00245 = 0.0006125$

Compute final weights of the network

Step 9

$V_{11}(new) = V_{11}(old) + \Delta V_{11} = 0.6 + 0 = 0.6$

$V_{12}(new) = V_{12}(old) + \Delta V_{12} = -0.3 + 0 = -0.3$

$V_{21}(new) = V_{21}(old) + \Delta V_{21} = -0.1 + 0.00295$
$= -0.09705$

$V_{22}(new) = 0.4006125$

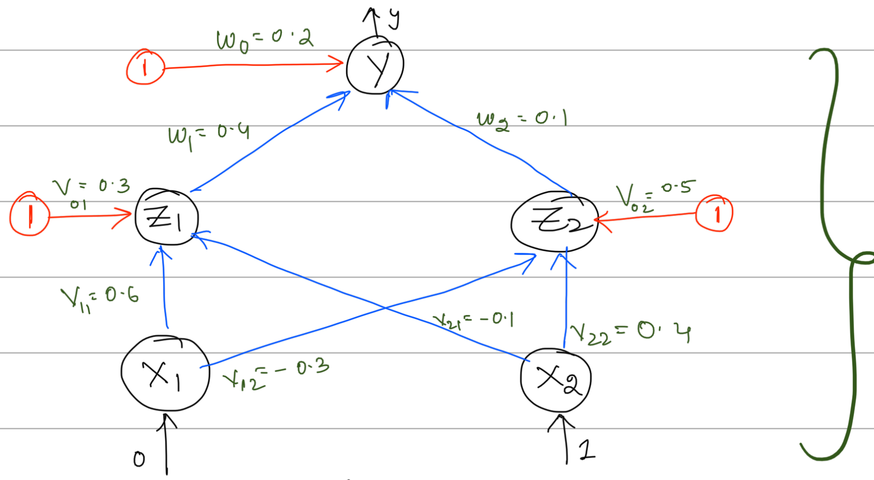
$w_1(new) = w_1(old) + \Delta w_1 = 0.4 + 0.0164 = 0.4164$

$w_2 (new) = 0.12117$

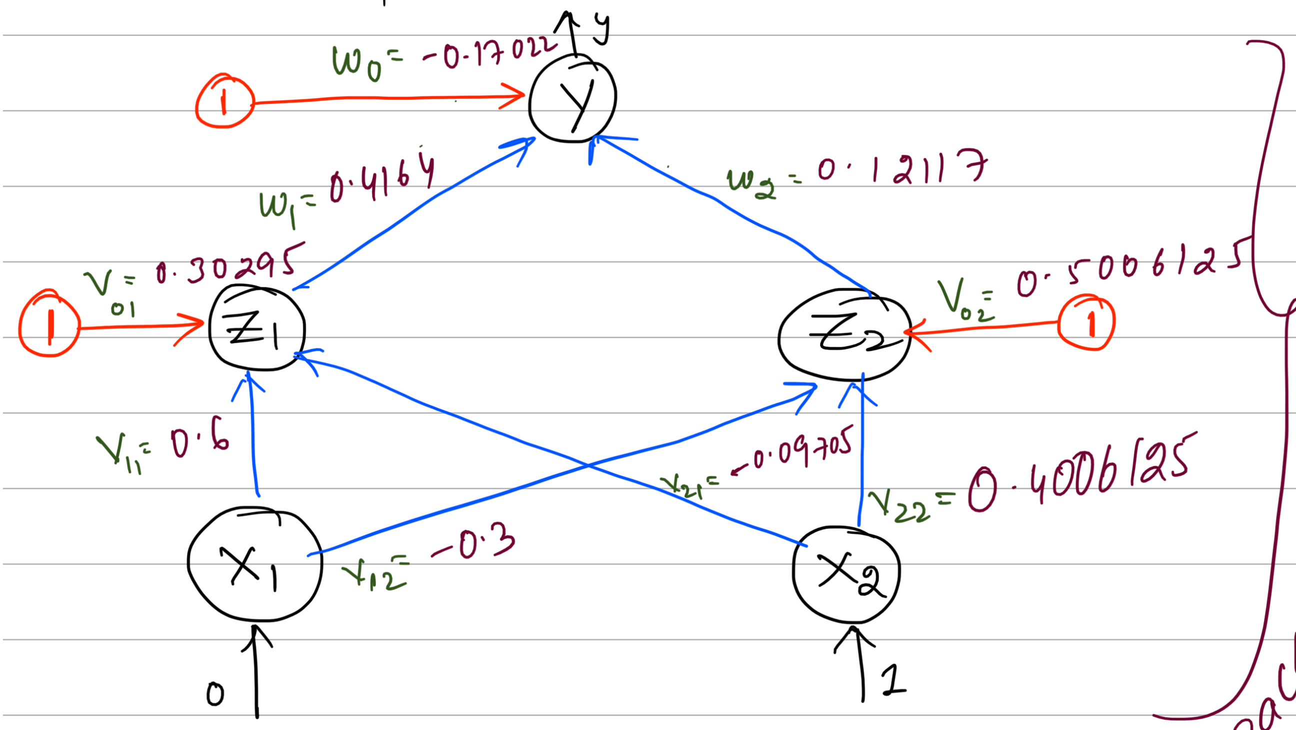
$V_{01} (new) = 0.30295$

$V_{02} (new) = 0.5006125$

$W_0 (new) = -0.17022.$



Before Backpropagation



After Backpropagation

↳ change in weights to fit model better