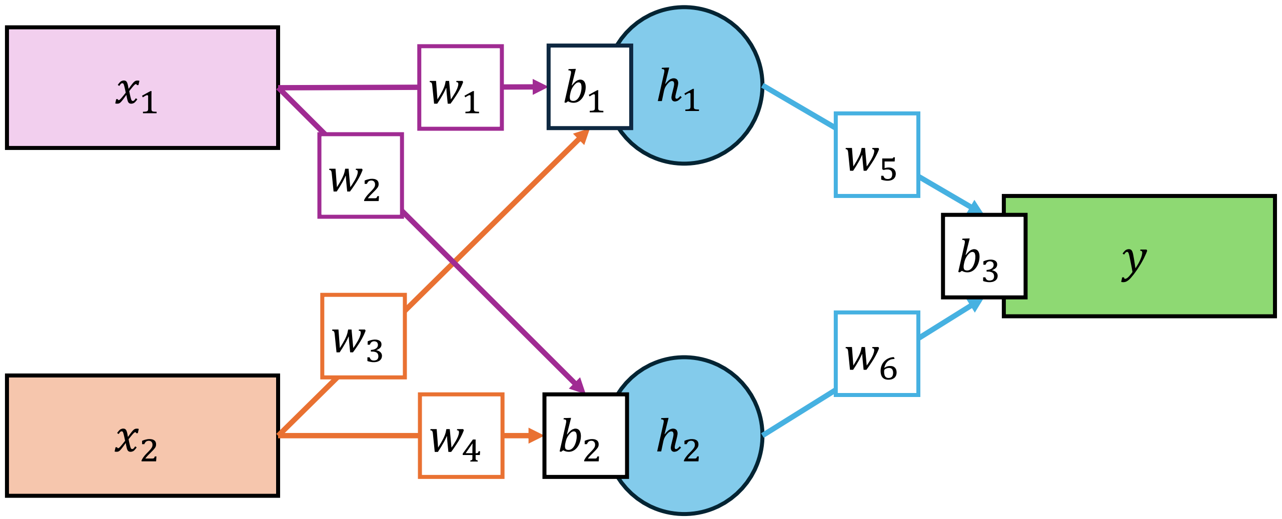
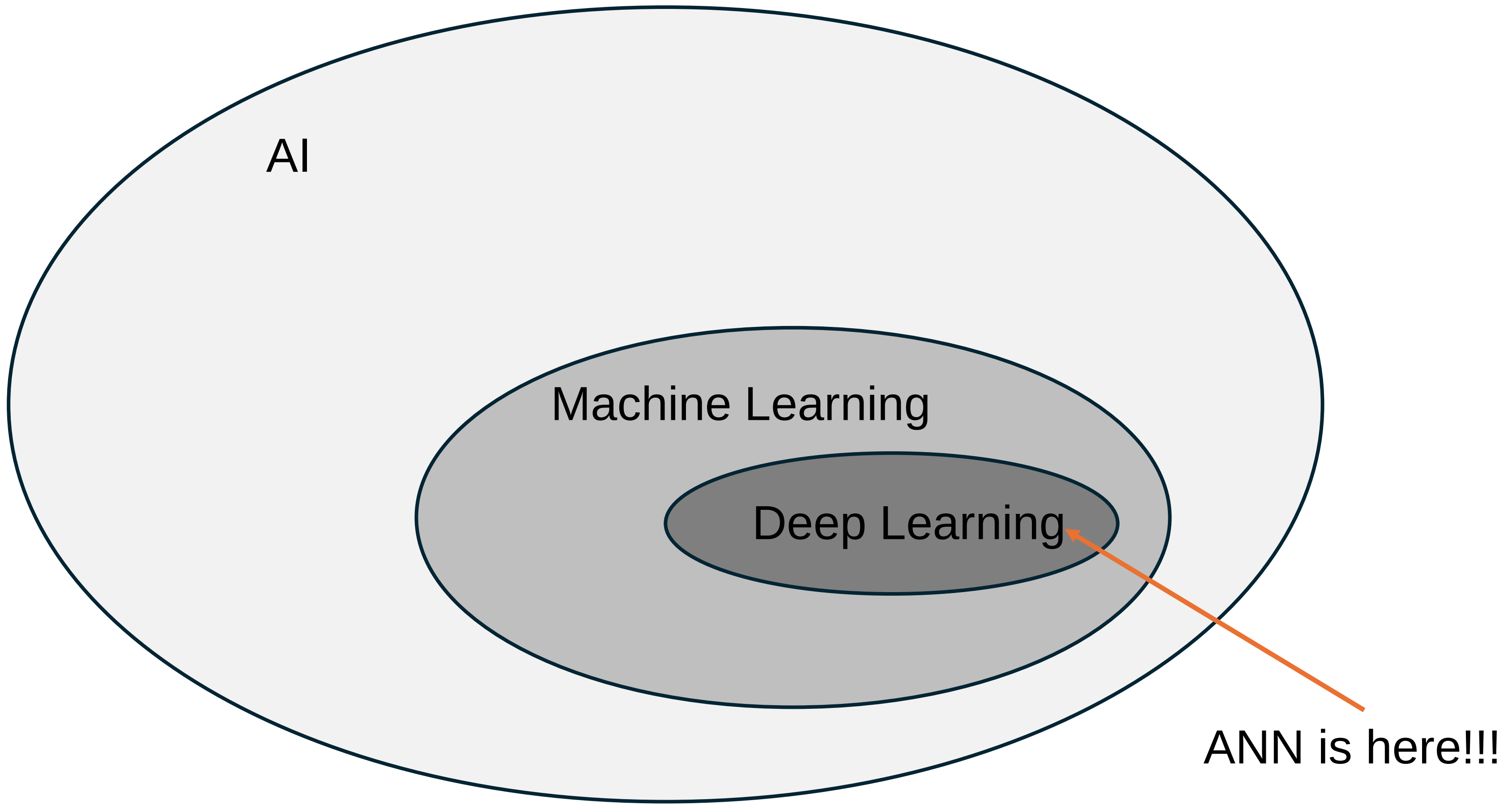
Artificial Neural Network

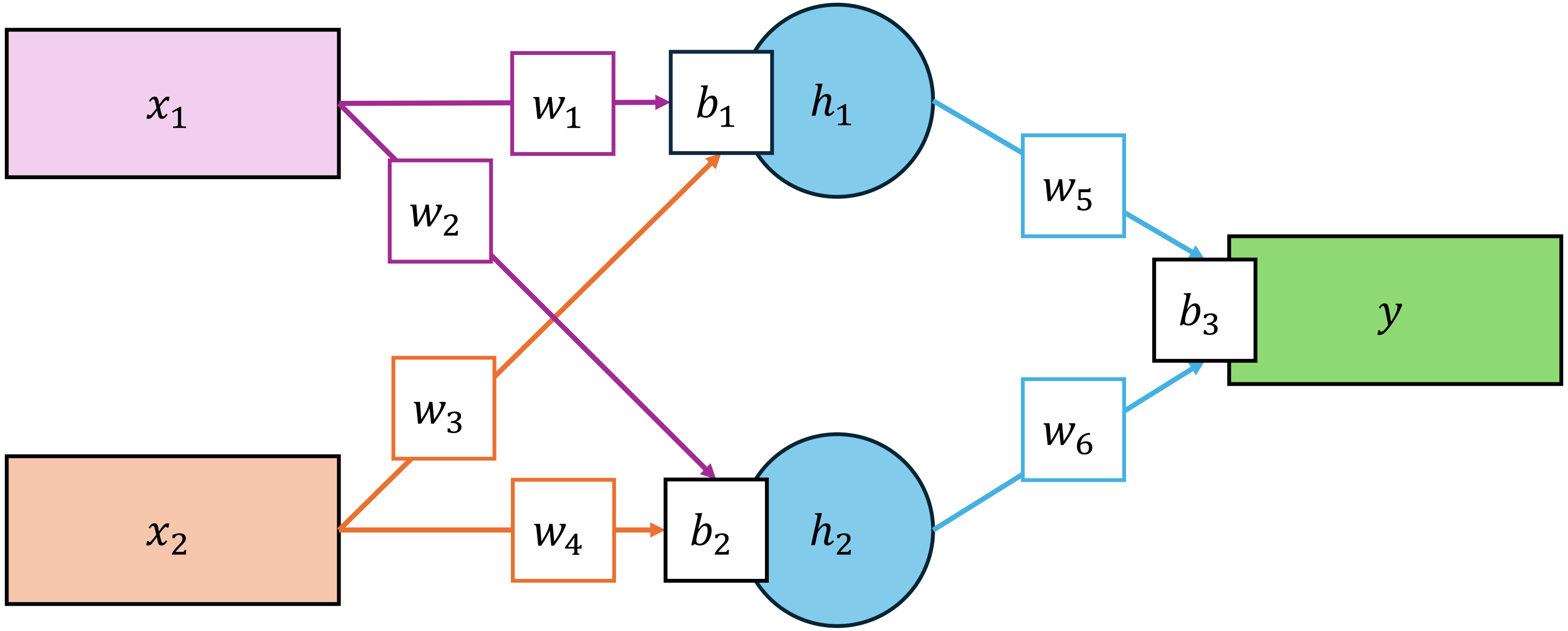
AKA. ANN

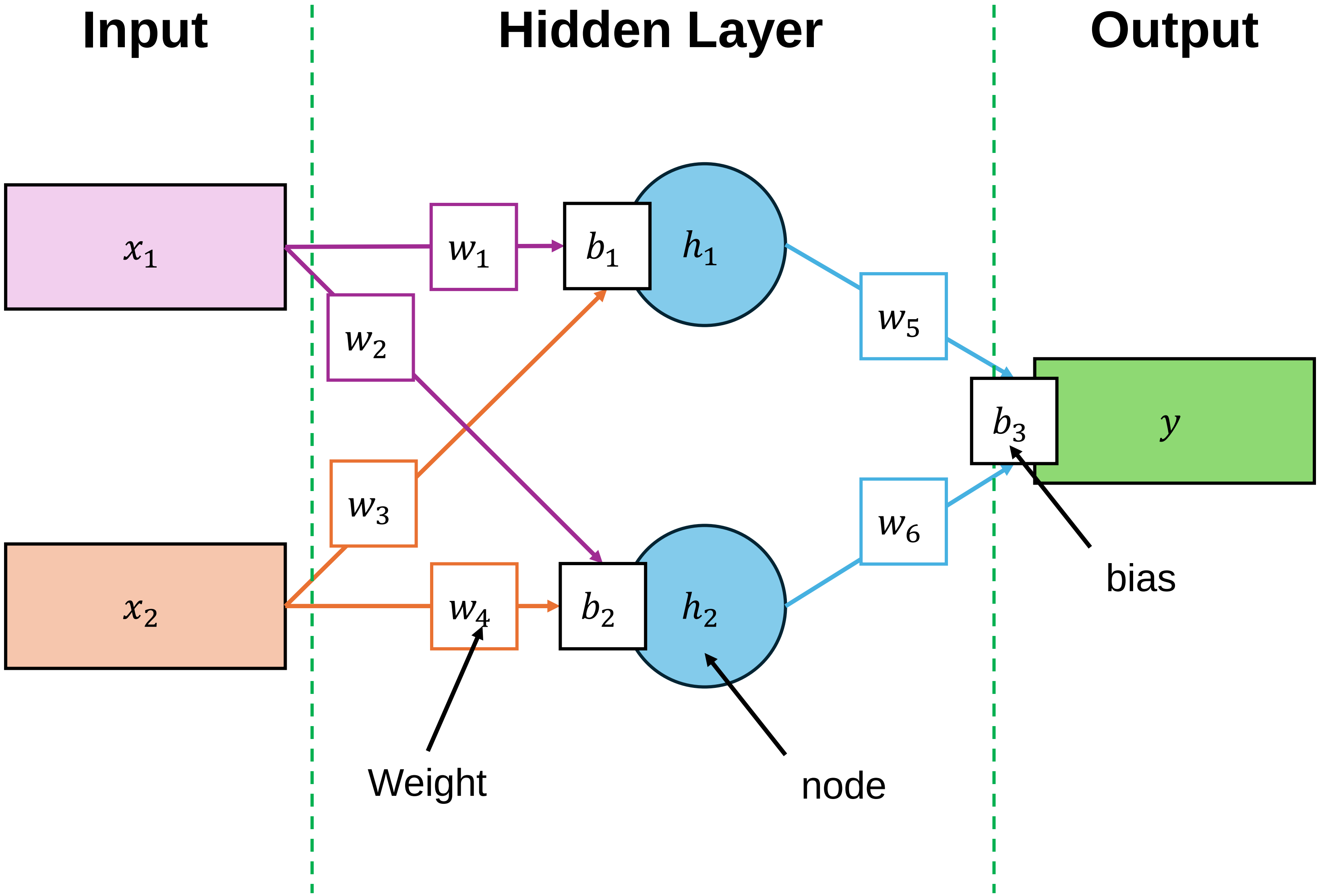


What artificial neural network is ???

A neural network is a model inspired by the structure and function of biological neural networks in the brains of living organisms.


AI
Machine Learning
Deep Learning
ANN is here!!!


x_1
w_1
b_1
h_1
w_2
w_5
b_3
y
w_3
w_6
x_2
w_4
b_2
h_2


Input
Hidden Layer
Output
x_1
x_2
w_1
w_2
w_3
w_4
w_5
w_6
b_1
b_2
b_3
h_1
h_2
y
Weight
node
bias

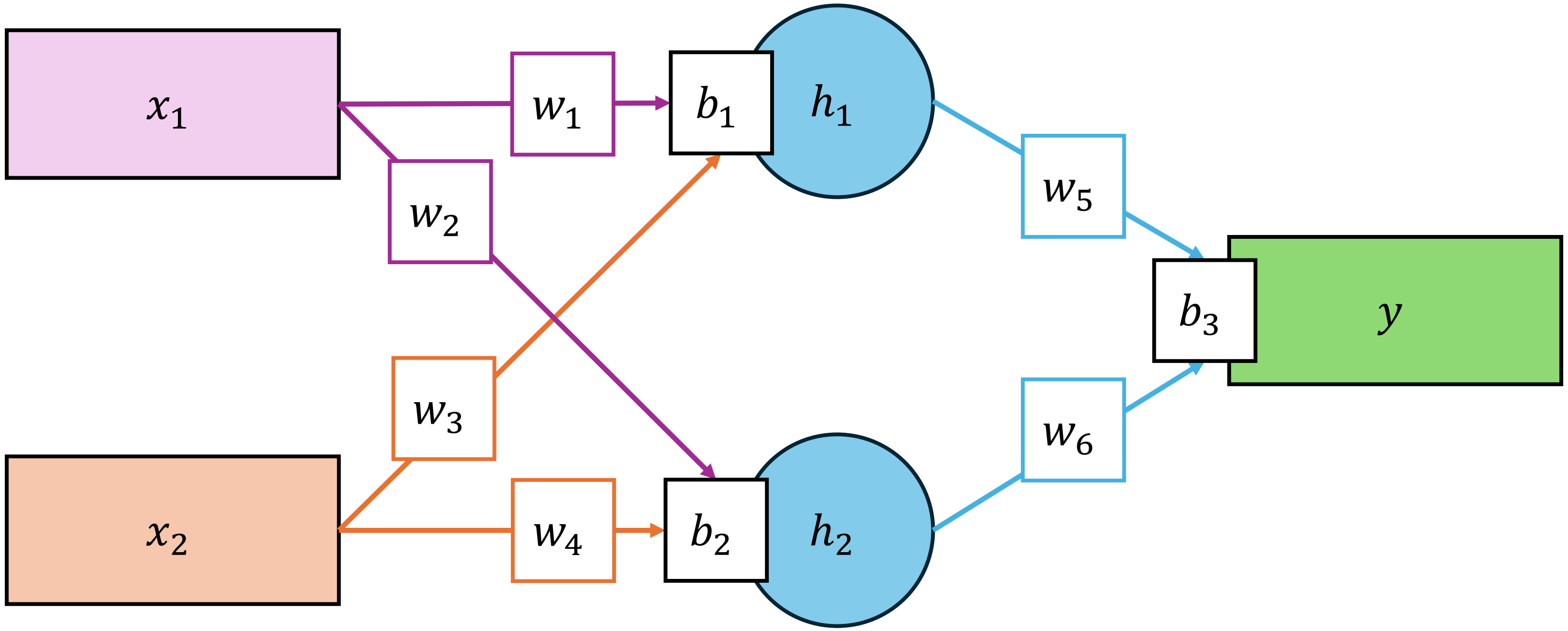
So far so good, How we can apply the ANN in real life ???

Let’s Say, you want to predict the value of the lottery.

From The Historical Data of Lottery

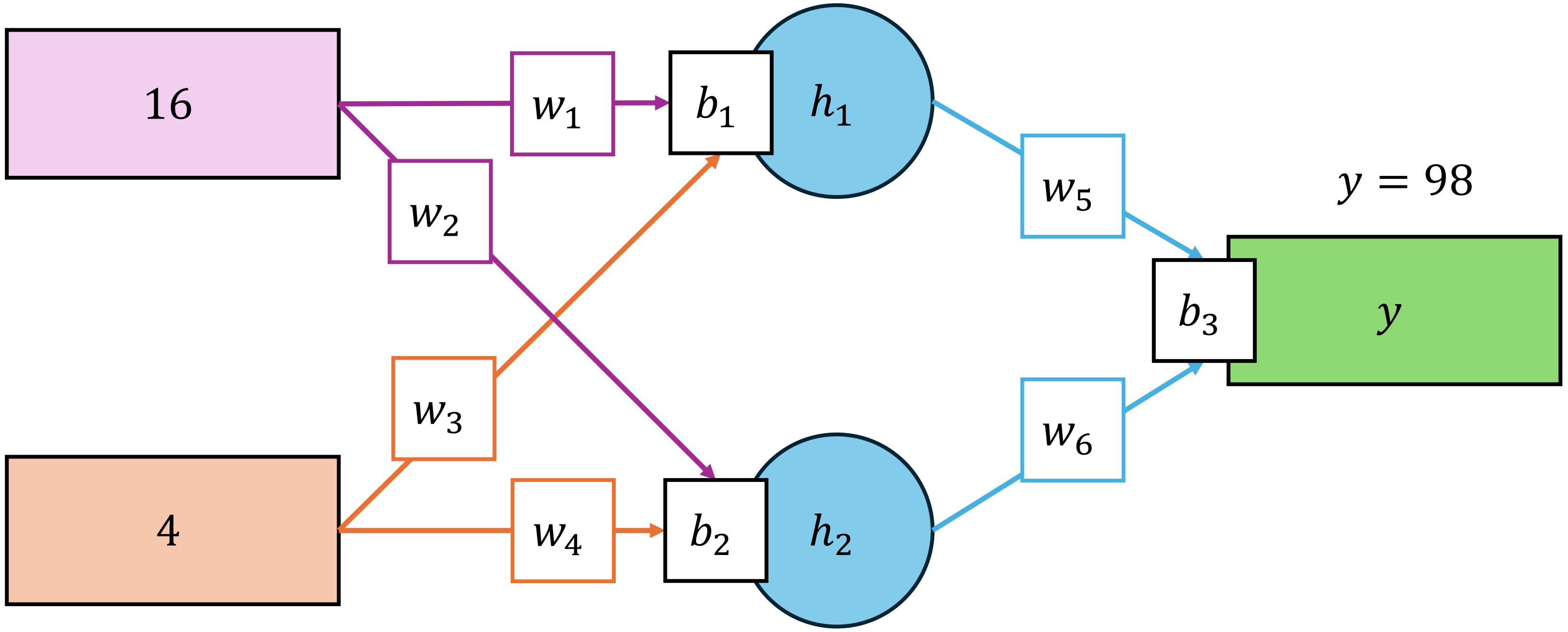
x_1(Date)	x_2(Month)	y(2nd Last Digit)
16	4	98
1	4	81
16	3	26

We're going to build a model that can accurately predict the 2nd last digit of a lottery number.



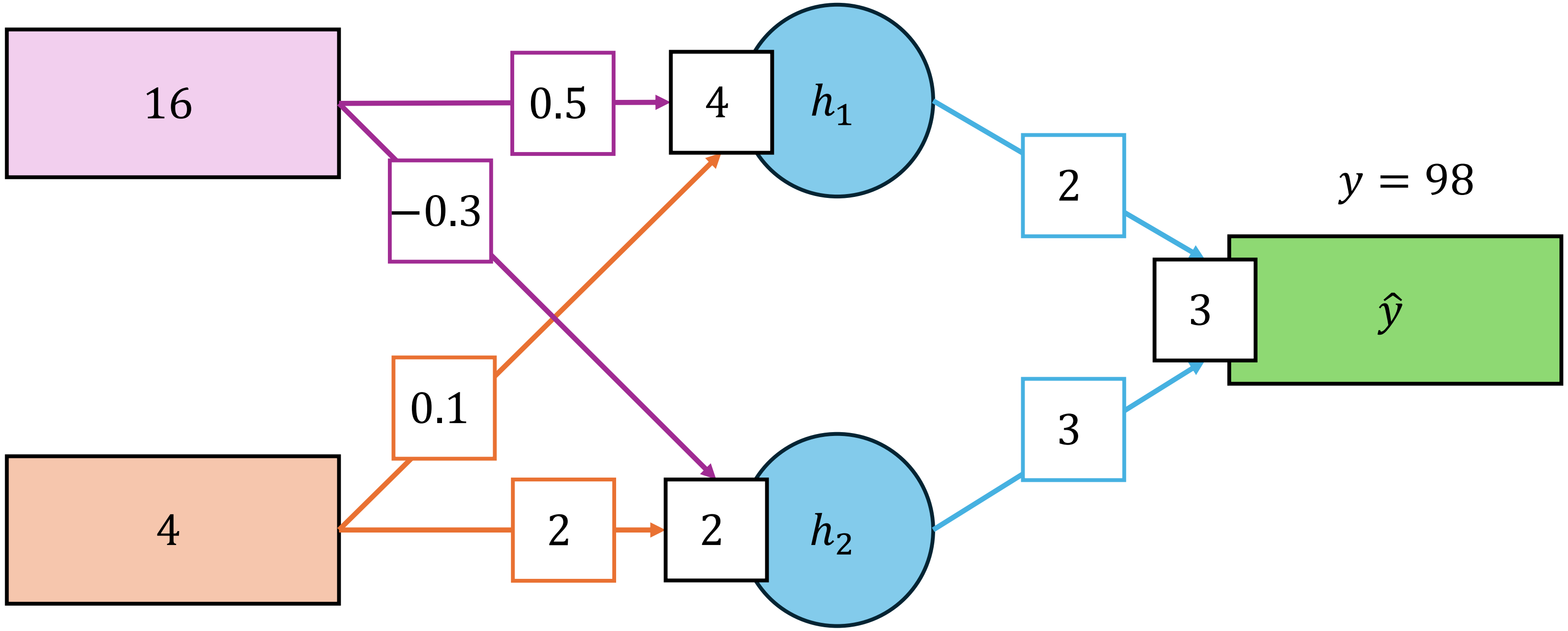
Let’s do Forward Propagation

Iteration: 1, **Epoch**: 1



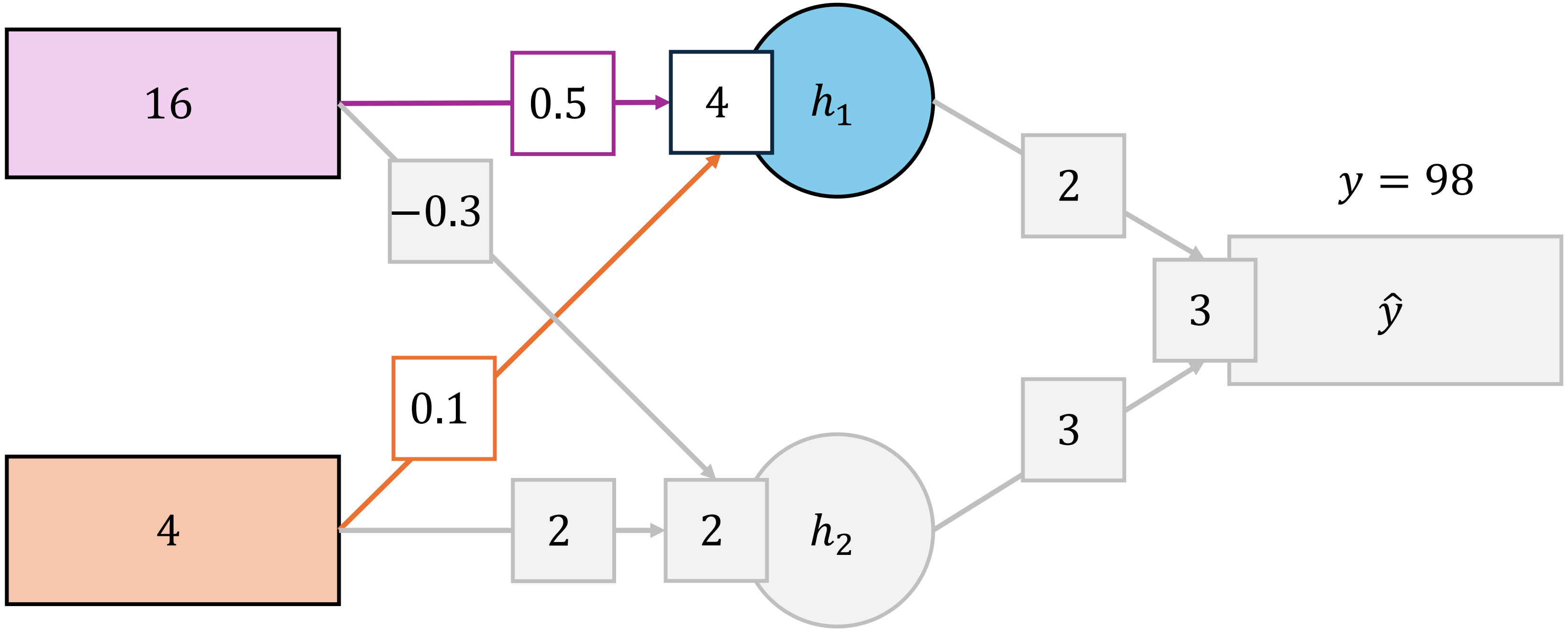
First at first, Just random the weight and bias.

Iteration: 1, **Epoch**: 1



First at first, Just random the weight and bias.

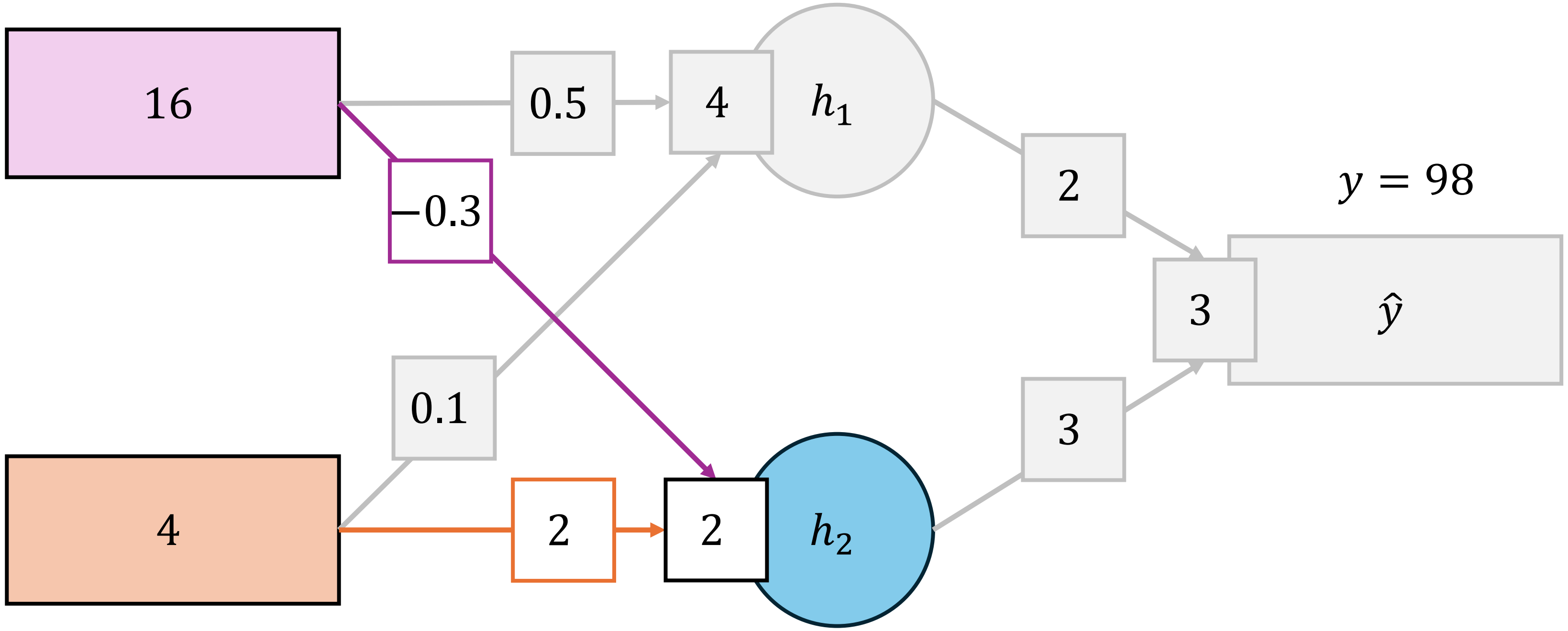
Iteration: 1, **Epoch**: 1



$$h_1 = (x_1 w_1 + x_2 w_3) + b_1$$

$$h_1 = ((16)(0.5) + (4)(0.1)\ ) + 4 = 12.4$$

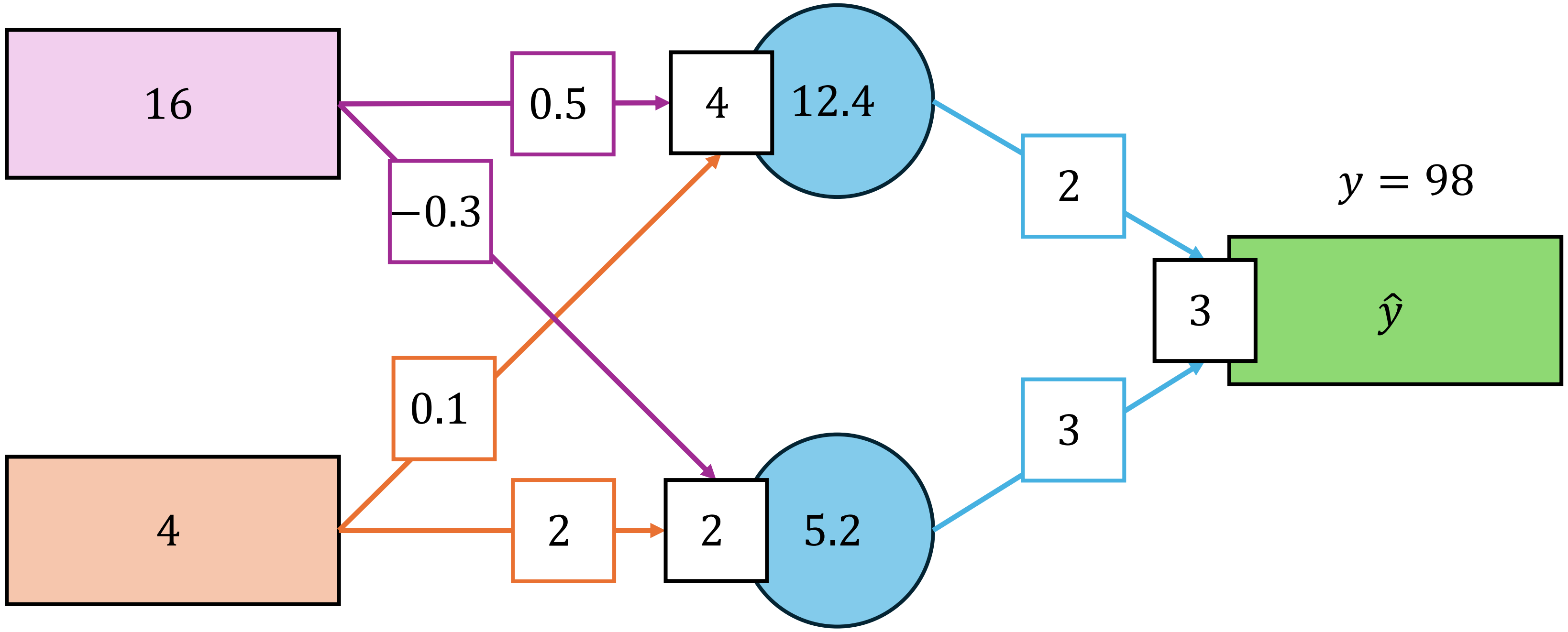
Iteration: 1, **Epoch**: 1



$$h_2 = (x_1 w_2 + x_2 w_4) + b_2$$

$$h_2 = ((16)(-0.3) + (4)(2)) + 2 = 5.2$$

Iteration: 1, **Epoch**: 1

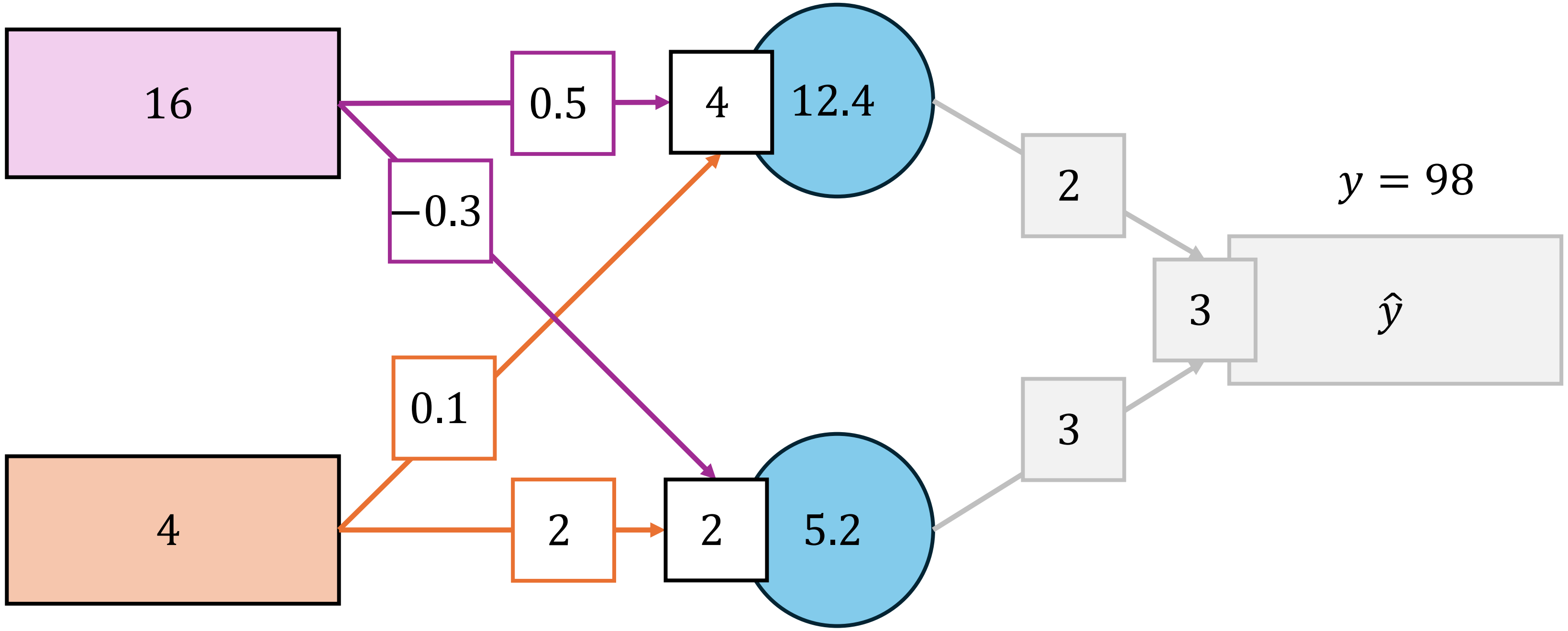


$h_1 = 12.4$

$h_2 = 5.2$

What if we use matrix for calculation ???

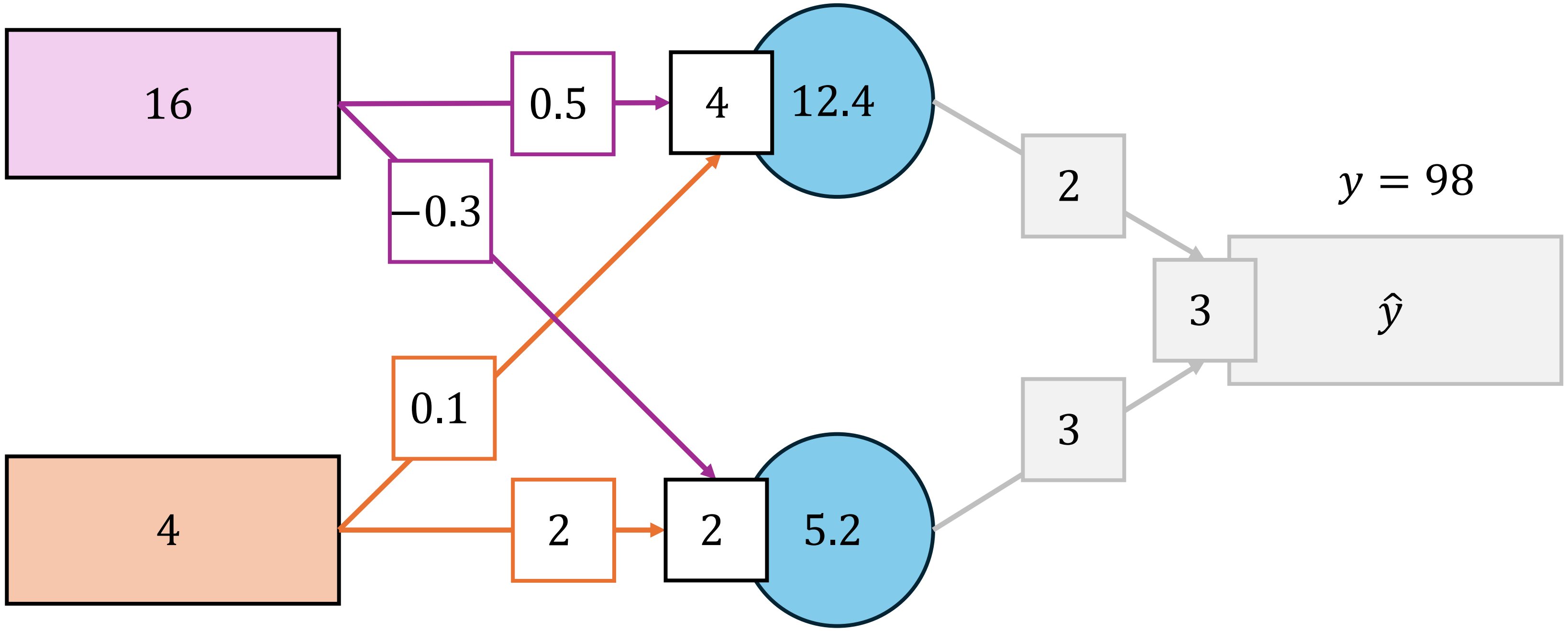
Iteration: 1, **Epoch**: 1



$$[h_1 \quad h_2] = [x_1 \quad x_2] \times \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} + [b_1 \quad b_2]$$

$$[h_1 \quad h_2] = [(x_1 w_1 + x_2 w_3) \quad (x_1 w_2 + x_2 w_4)] + [b_1 \quad b_2]$$

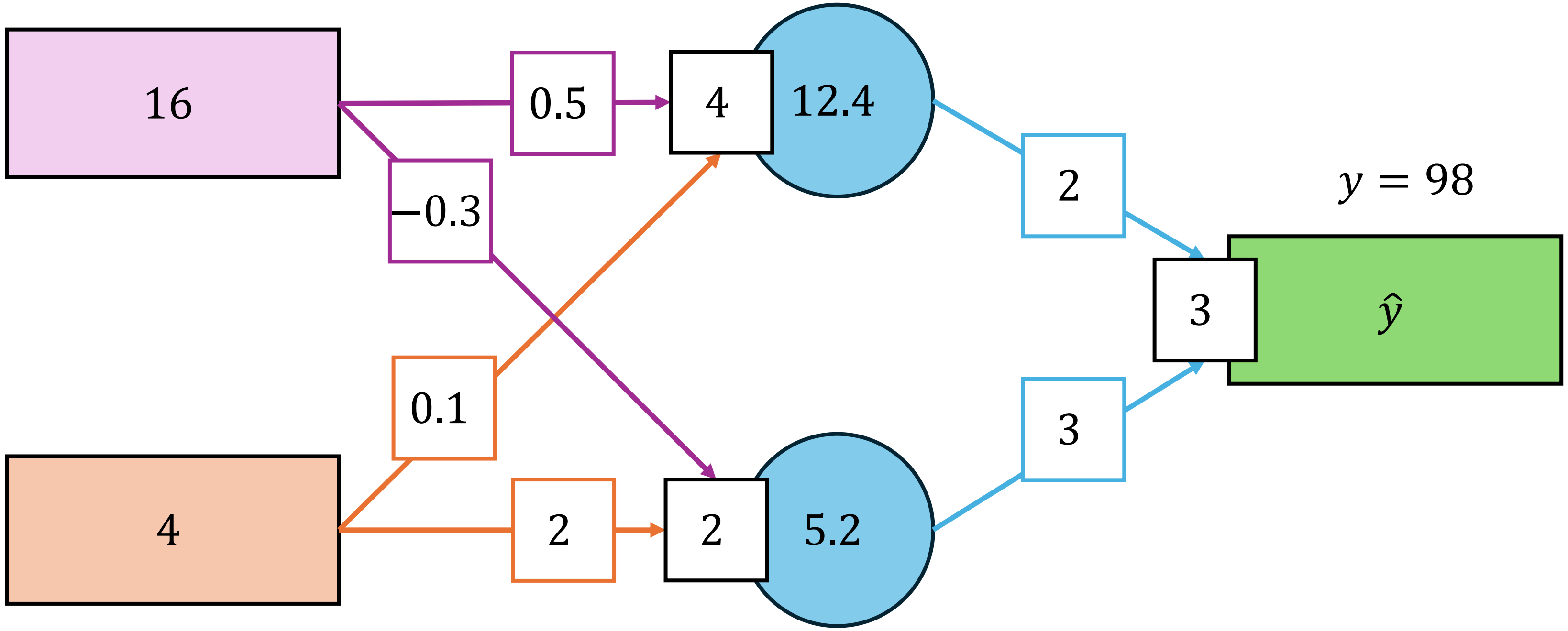
Iteration: 1, **Epoch**: 1



$$[h_1 \quad h_2] = [((16)(0.5) + (4)(0.1)\,) \quad ((16)(-0.3) + (4)(2)\,)] + [4 \quad 2]$$

$$[h_1 \quad h_2] = [12.4 \quad 5.2]$$

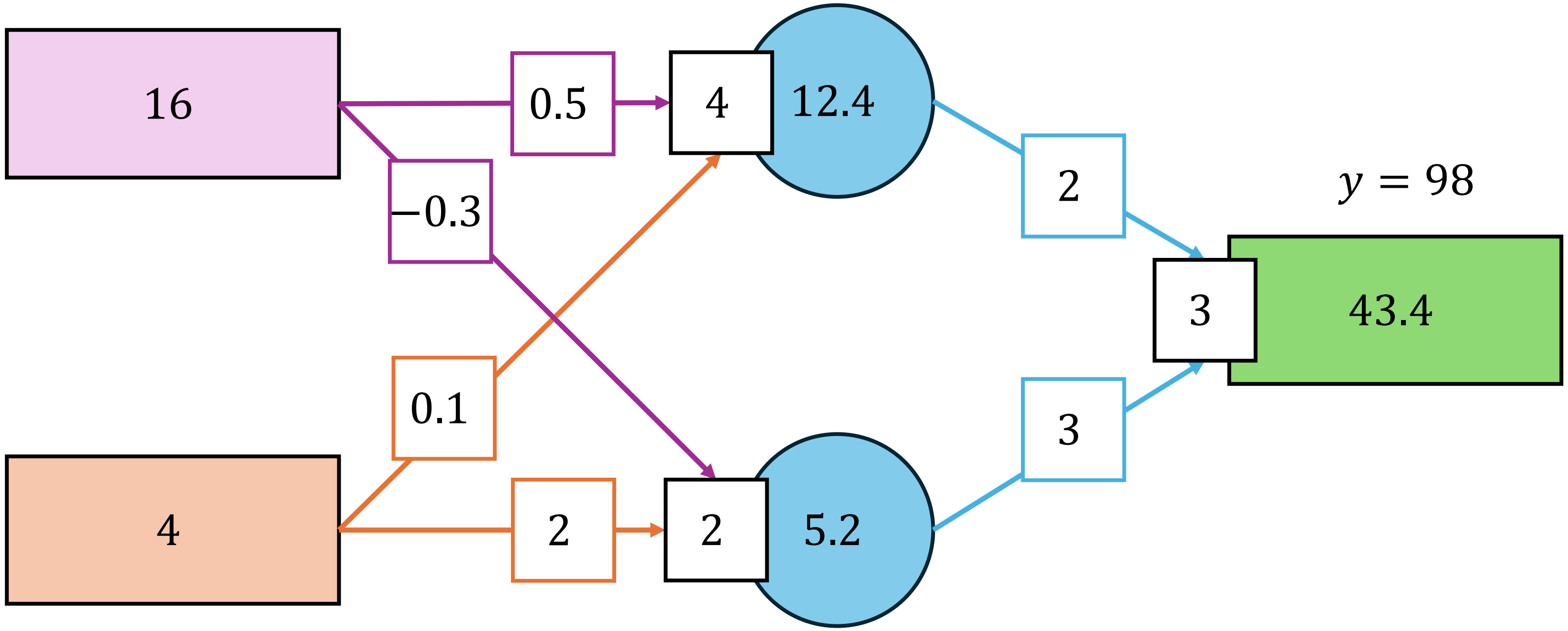
Iteration: 1, **Epoch**: 1



$$[\hat{y}] = [h_1 \quad h_2] \times \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} + [b_3]$$

$$[\hat{y}] = [12.4 \quad 5.2] \times \begin{bmatrix} 2 \\ 3 \end{bmatrix} + [3] = [43.4]$$

Iteration: 1, **Epoch**: 1



$$[\hat{y}] = [h_1 \quad h_2] \times \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} + [b_3]$$

$$[\hat{y}] = [12.4 \quad 5.2] \times \begin{bmatrix} 2 \\ 3 \end{bmatrix} + [3] = [43.4]$$

Let’ calculate the loss

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$Where: N\ is\ numbers\ of\ train\ data$

Then, **N = 3**

x_1(Date)	x_2(Month)	y(2nd Last Digit)
16	4	98
1	4	81
16	3	26

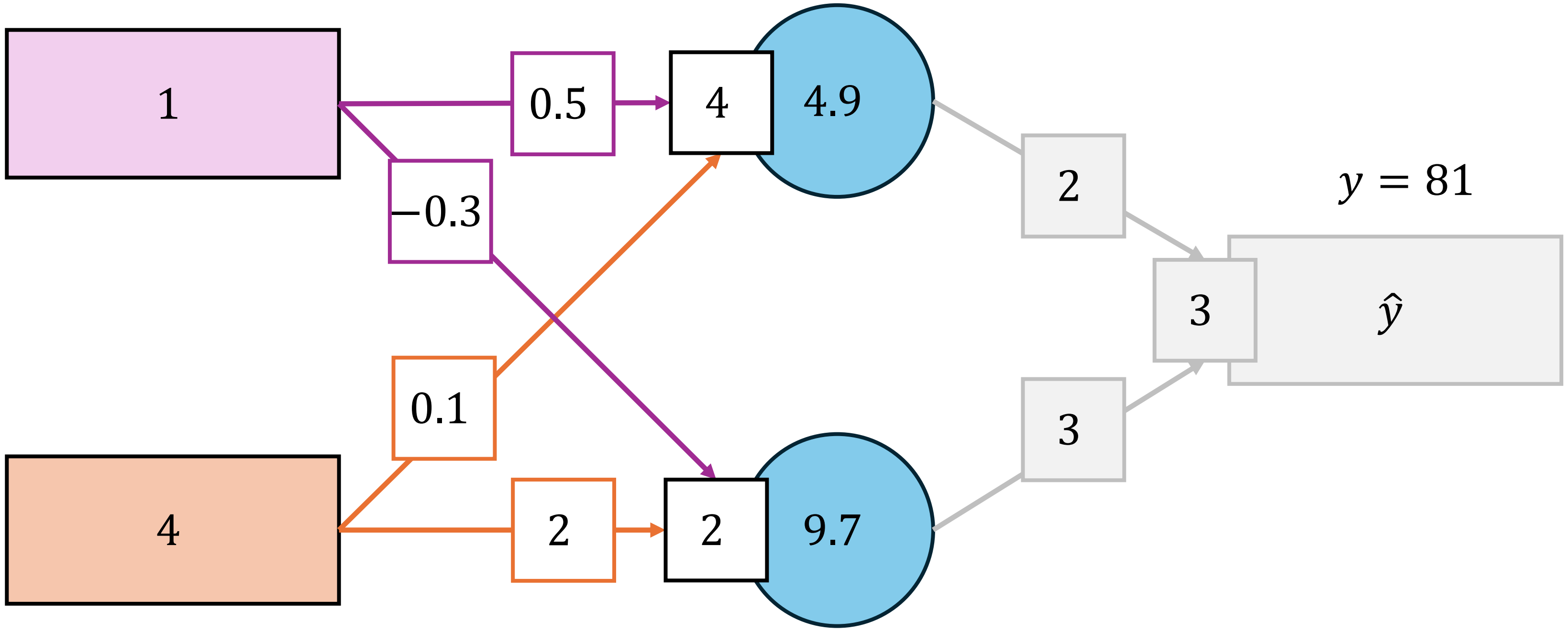
Iteration: 1, **Epoch**: 1

i	x_1	x_2	y	h_1	h_2	$\hat{y}$	$(y_i - \hat{y}_i)$
1	16	4	98	12.4	5.4	43.4	54.6
2	1	4	81				
3	16	3	26				

$$Error = (y_1 - \hat{y}_1)^2 = ((98) - (43.4))^2 = 2981.16$$

End of Iteration: 1 Epoch: 1

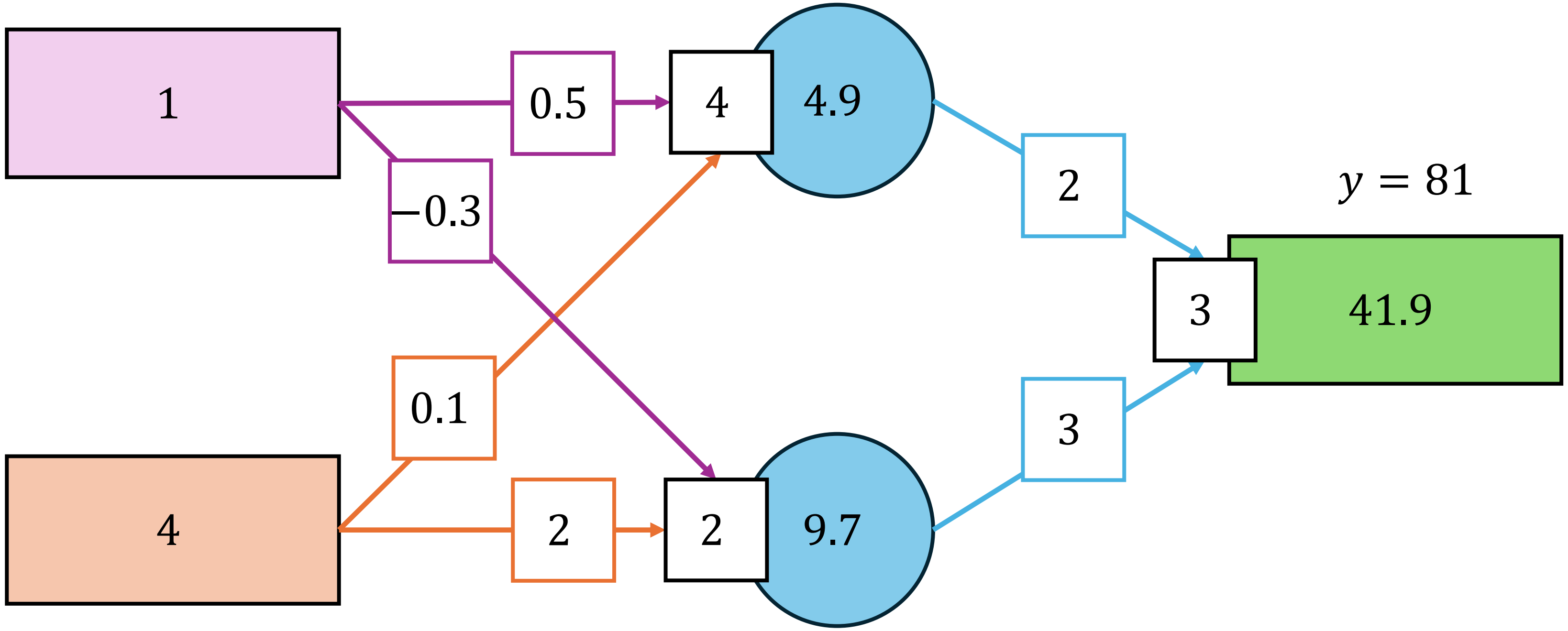
Iteration: 2, **Epoch**: 1



$$[h_1 \quad h_2] = [x_1 \quad x_2] \times \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} + [b_1 \quad b_2]$$

$$[h_1 \quad h_2] = [1 \quad 4] \times \begin{bmatrix} 0.5 & -0.3 \\ 0.1 & 4 \end{bmatrix} + [4 \quad 2] = [4.9 \quad 9.7]$$

Iteration: 2, **Epoch**: 1



$$[\hat{y}] = [h_1 \quad h_2] \times \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} + [b_3]$$

$$[\hat{y}] = [4.9 \quad 9.7] \times \begin{bmatrix} 2 \\ 3 \end{bmatrix} + [3] = [41.9]$$

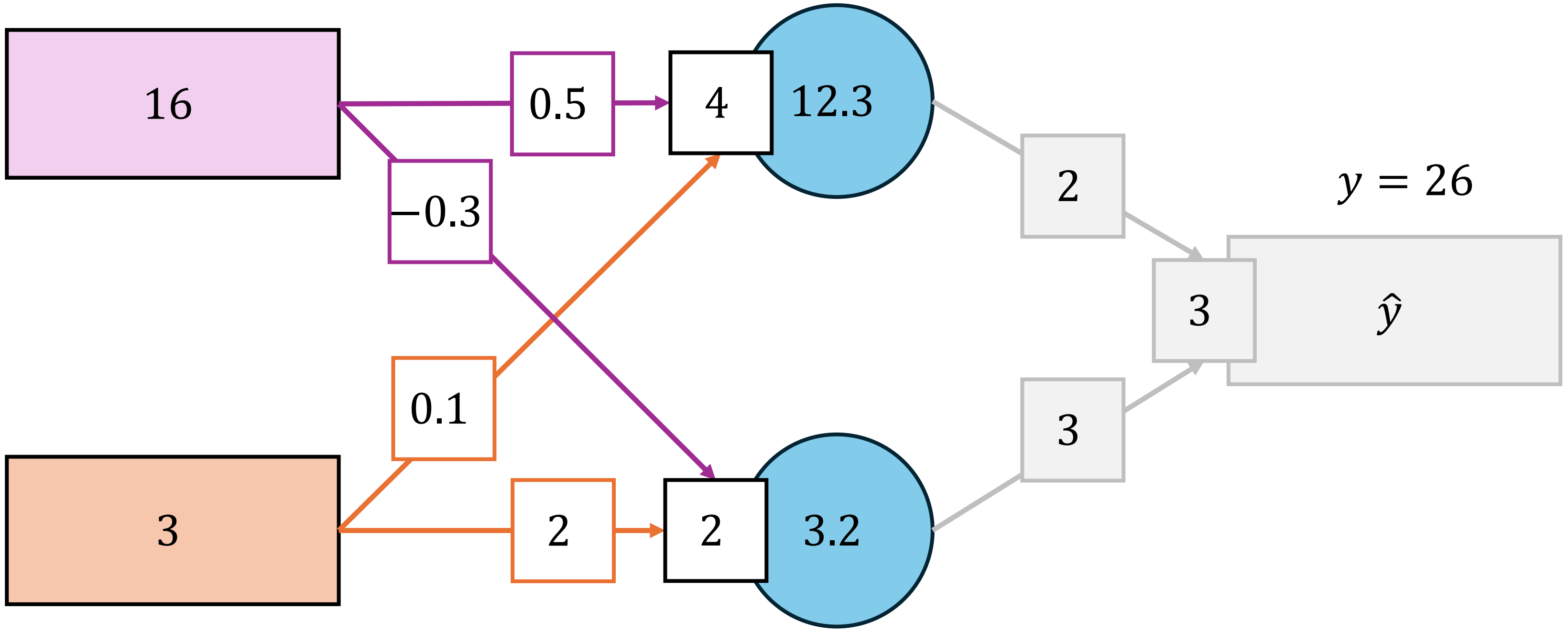
Iteration: 2, **Epoch**: 1

i	x_1	x_2	y	h_1	h_2	$\hat{y}$	$(y_i - \hat{y}_i)$
1	16	4	98	12.4	5.4	43.4	54.6
2	1	4	81	4.9	9.7	41.9	39.1
3	16	3	26				

$$Error = (y_2 - \hat{y}_2)^2 = ((81) - (41.9))^2 = 1528.81$$

End of Iteration: 2 Epoch: 1

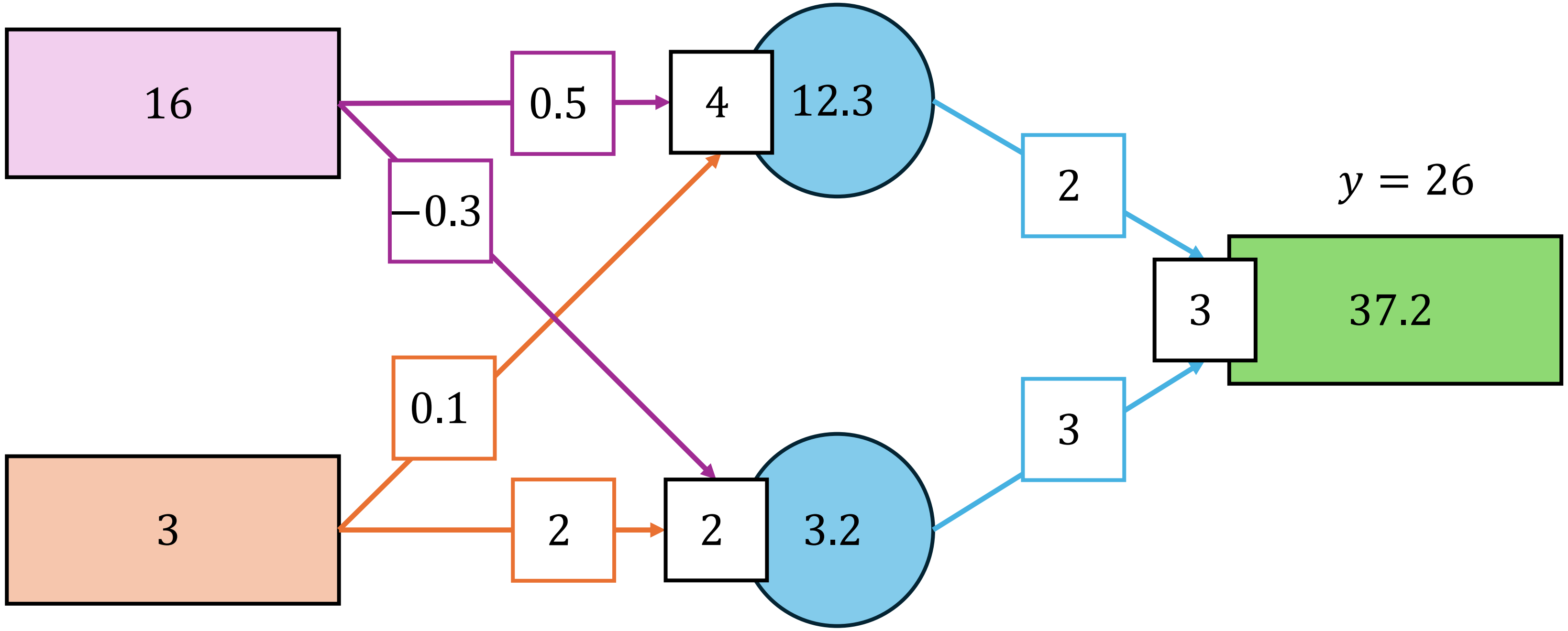
Iteration: 3, **Epoch**: 1



$$[h_1 \quad h_2] = [x_1 \quad x_2] \times \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} + [b_1 \quad b_2]$$

$$[h_1 \quad h_2] = [16 \quad 3] \times \begin{bmatrix} 0.5 & -0.3 \\ 0.1 & 4 \end{bmatrix} + [4 \quad 2] = [12.3 \quad 3.2]$$

Iteration: 3, **Epoch**: 1



$$[\hat{y}] = [h_1 \quad h_2] \times \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} + [b_3]$$

$$[\hat{y}] = [12.3 \quad 3.2] \times \begin{bmatrix} 2 \\ 3 \end{bmatrix} + [3] = [41.9]$$

Iteration: 3, **Epoch**: 1

i	x_1	x_2	y	h_1	h_2	$\hat{y}$	$(y_i - \hat{y}_i)$
1	16	4	98	12.4	5.4	43.4	54.6
2	1	4	81	4.9	9.7	41.9	39.1
3	16	3	26	12.3	3.2	37.2	-11.2

$$Error = (y_3 - \hat{y}_3)^2 = ((26) - (37.2))^2 = 125.44$$

End of Iteration: 3 Epoch: 1

Time to calculate total loss and do **Backpropagation**

i	x_1	x_2	y	h_1	h_2	$\hat{y}$	$(y_i - \hat{y}_i)$
1	16	4	98	12.4	5.4	43.4	54.6
2	1	4	81	4.9	9.7	41.9	39.1
3	16	3	26	12.3	3.2	37.2	-11.2

Total Loss

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$MSE = \frac{1}{3}(54.6^2 + 39.1^2 + (-11.2)^2) = 4635.41$$

Backpropagation

We’re going to use **Gradient Descent** to **minimize the MSE** and **update all weights and bias**.

Apply Gradient Descent

$$w_{new} = w_{current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w}$$

For weights

$$b_{new} = b_{current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b}$$

For biases

$Given$: $\alpha = 0.001$

$$w_{new} = w_{current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w}$$

$$w_{new} = w_{current} - \alpha \left(\frac{\partial MSE_1}{\partial w} + \frac{\partial MSE_2}{\partial w} + \frac{\partial MSE_3}{\partial w} \right)$$

$$b_{new} = b_{current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b}$$

$$b_{new} = b_{current} - \alpha \left(\frac{\partial MSE_1}{\partial b} + \frac{\partial MSE_2}{\partial b} + \frac{\partial MSE_3}{\partial b} \right)$$

For weights

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w} = \begin{bmatrix} \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_1} \\ \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_2} \\ \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_3} \\ \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_4} \\ \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_5} \\ \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_6} \end{bmatrix}$$

For biases

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b} = \begin{bmatrix} \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_1} \\ \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_2} \\ \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_3} \end{bmatrix}$$

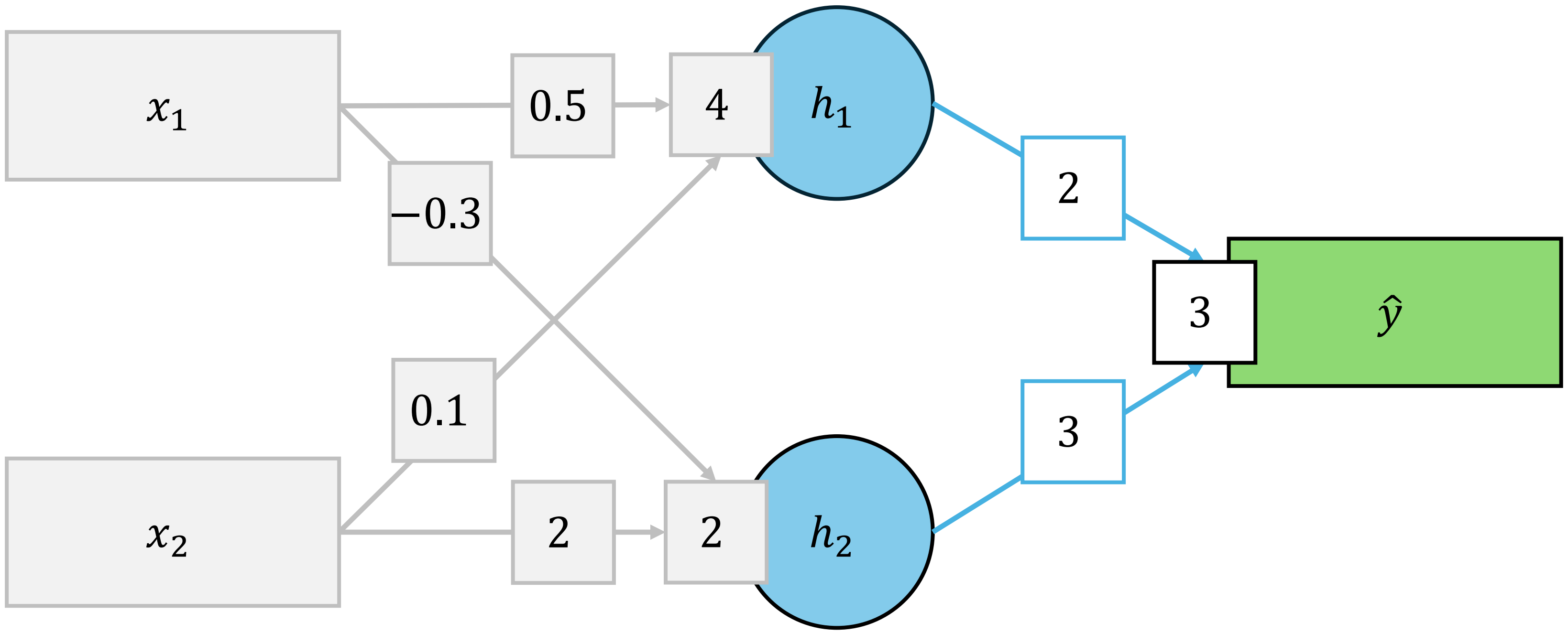
$$w_{new} = w_{current} - \alpha \boxed{\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w}}$$

$$b_{new} = b_{current} - \alpha \boxed{\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b}}$$

Find this!!!

$$Given: \alpha = 0.001$$

w_5



$$w_{5new} = w_{5current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_5}$$

Apply Chain Rule!!!

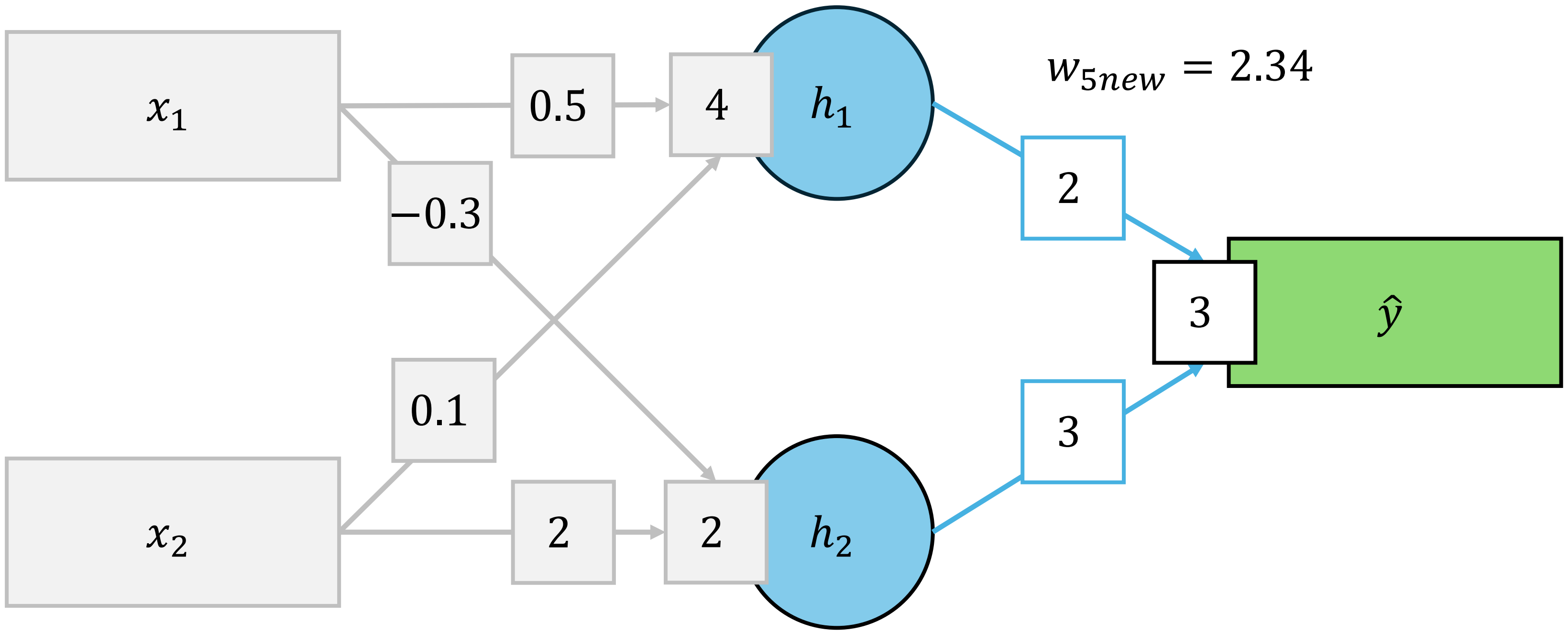
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_5} = \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_5}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_5} = \frac{1}{3} \sum_{i=1}^{n} \frac{\partial (y_i - \hat{y}_i)^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_5 + h_2 w_6 + b_3)}{\partial w_5}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_5} = -\frac{2}{3} \sum_{i=1}^{n} (y_i - \hat{y}_i) \cdot h_1$$

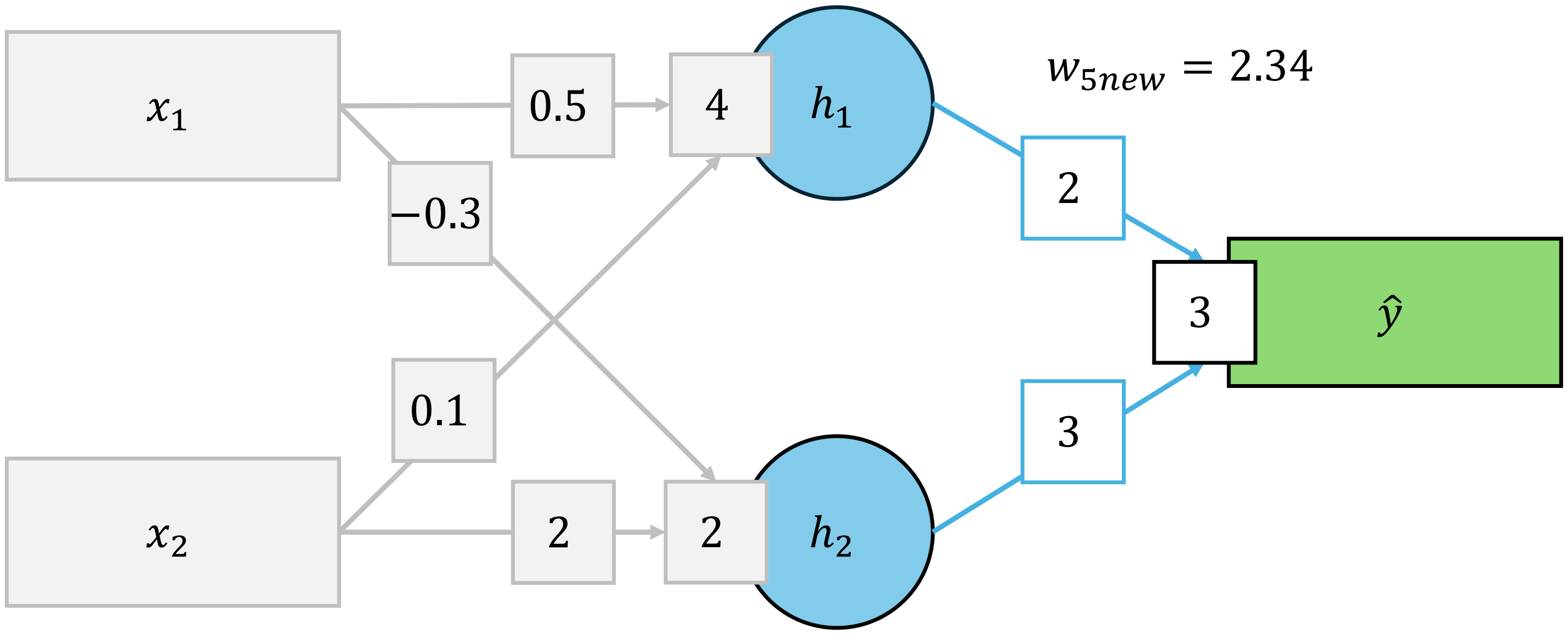
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_5} = -\frac{2}{3} [(54.6)(12.4) + (39.1)(4.9) + (-11.2)(8.3)] = -341.65$$

W_5



$$w_{5new} = 2 - (0.001)(-341.65) = 2.34$$

w_6



$$w_{6new} = w_{6current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_6}$$

Apply Chain Rule!!!

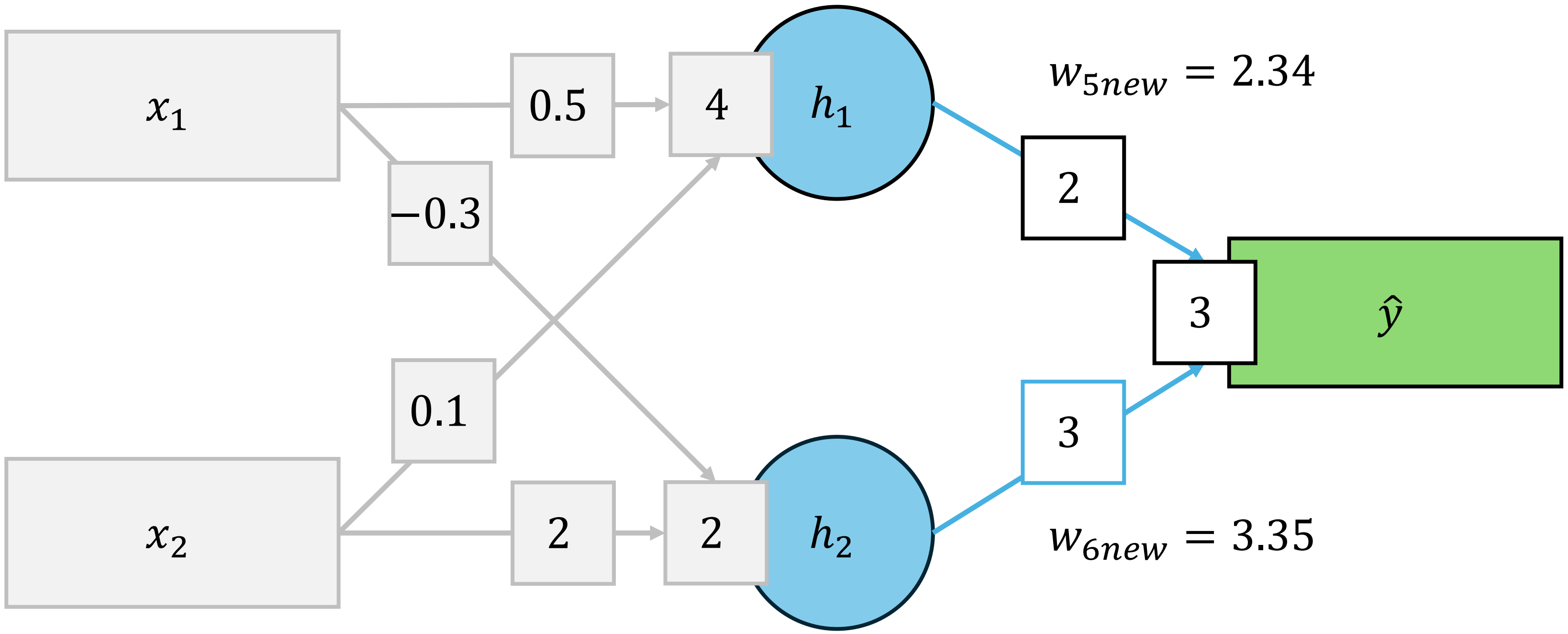
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_6} = \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_6}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_6} = \frac{1}{3} \sum_{i=1}^{n} \frac{\partial (y_i - \hat{y_i})^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_5 + h_2 w_6 + b_3)}{\partial w_6}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_6} = -\frac{2}{3} \sum_{i=1}^{n} (y_i - \hat{y_i}) \cdot h_2$$

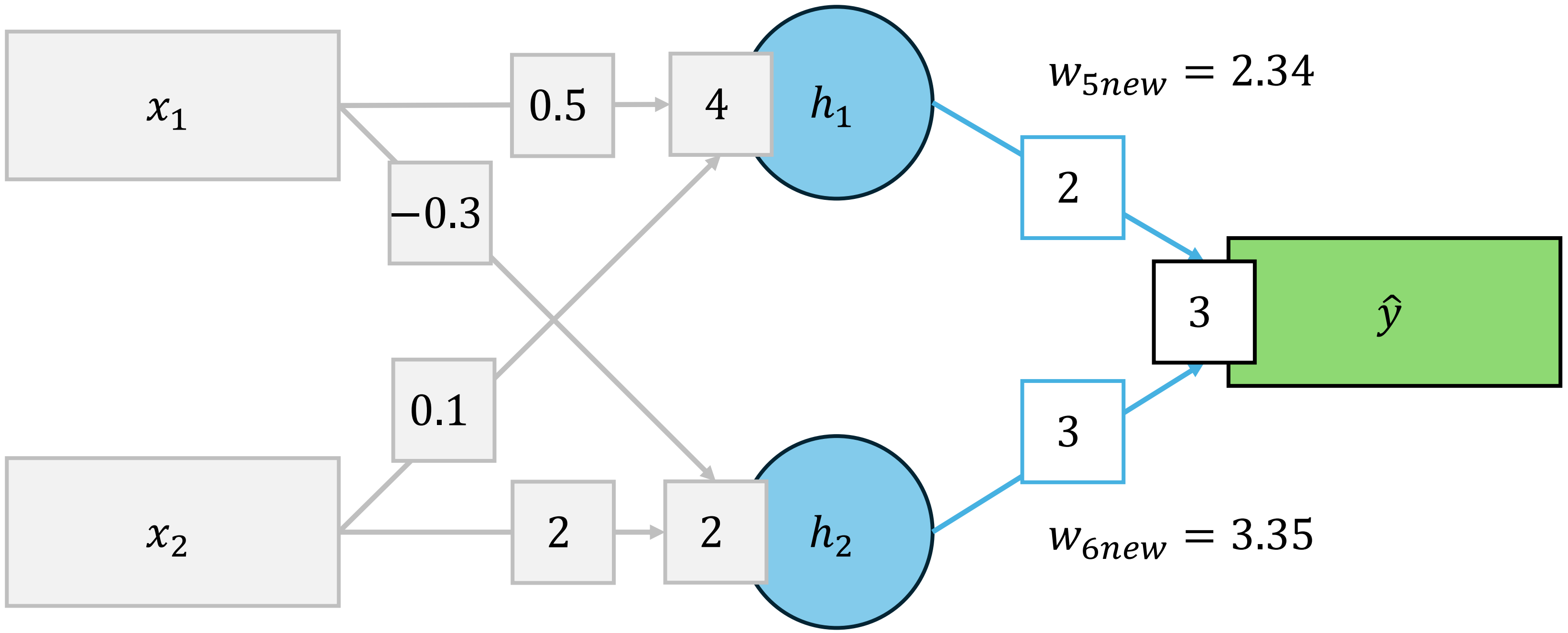
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_6} = -\frac{2}{3} [(54.6)(12.4) + (39.1)(4.9) + (-11.2)(8.3)] = -345.43$$

W_6



$$w_{6new} = 3 - (0.001)(-345.43) = 3.35$$

$\boldsymbol{b_3}$



$$b_{3new} = b_{3current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_3}$$

Apply Chain Rule!!!

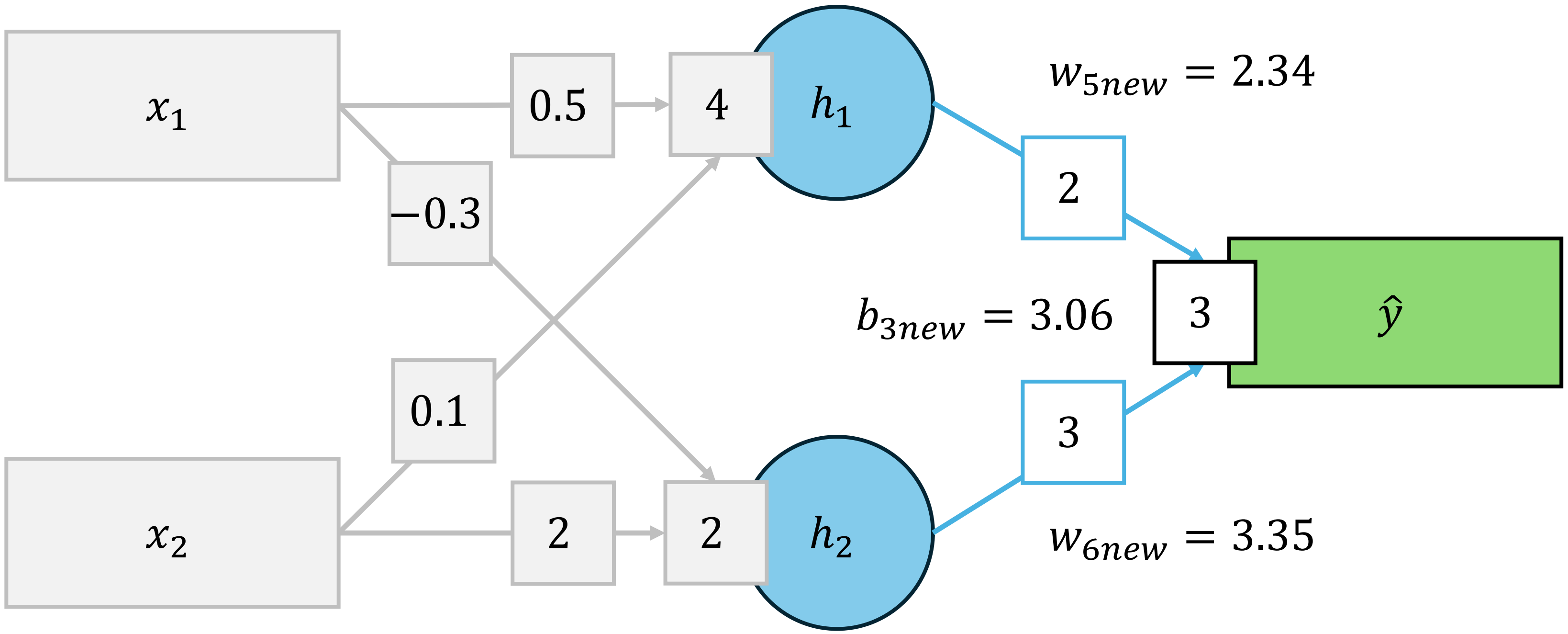
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_3} = \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b_3}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_3} = \frac{1}{3} \sum_{i=1}^{n} \frac{\partial (y_i - \widehat{y_i})^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_5 + h_2 w_6 + b_3)}{\partial b_3}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_3} = -\frac{2}{3} \sum_{i=1}^{n} (y_i - \widehat{y_i})$$

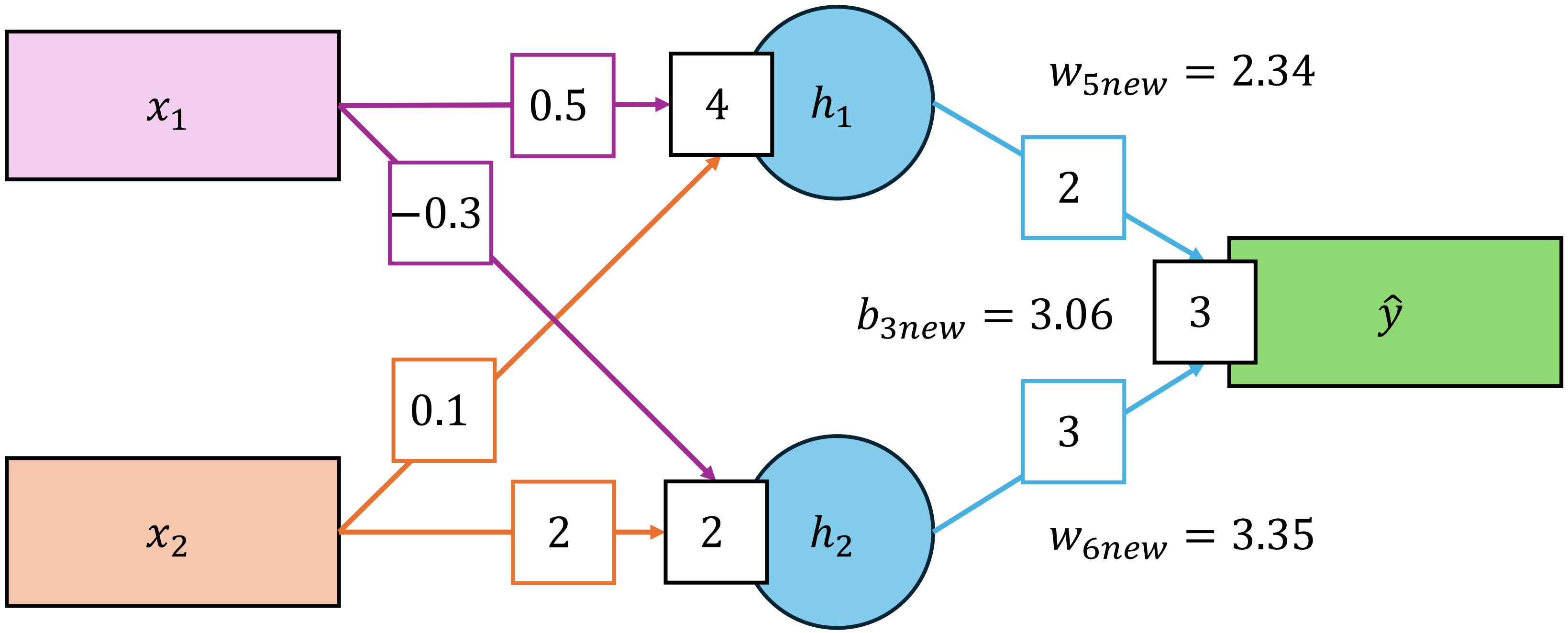
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_3} = -\frac{2}{3} [(54.6) + (39.1) + (-11.2)] = -55$$

b_3



$$b_{3new} = 3 - (0.001)(-55) = 3.06$$

w_1



$$w_{1new} = w_{1current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_1}$$

Apply Chain Rule!!!

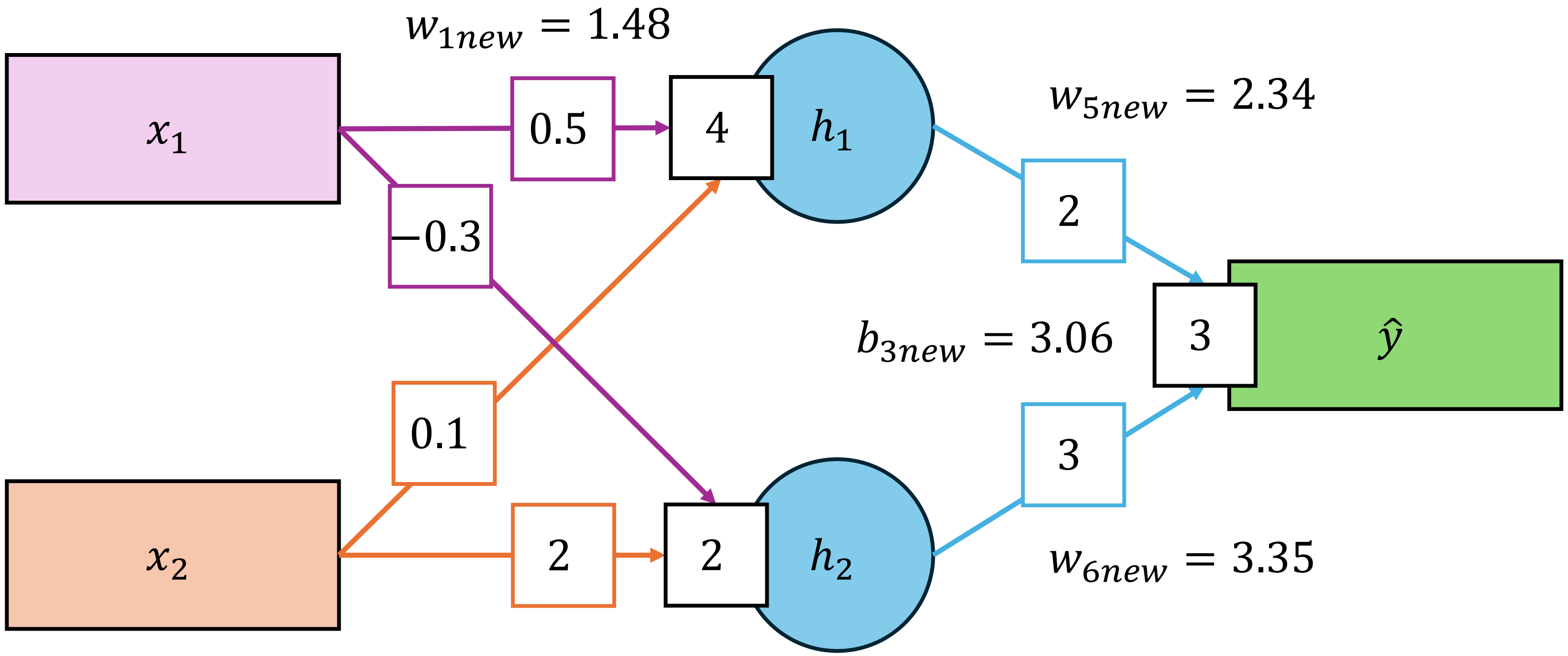
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_1} = \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_1}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_1} = \frac{1}{3} \sum_{i=1}^{n} \frac{\partial (y_i - \widehat{y_i})^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_5 + h_2 w_6 + b_3)}{\partial h_1} \cdot \frac{\partial (x_1 w_1 + x_2 w_3 + b_1)}{\partial w_1}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_1} = -\frac{2}{3} \cdot w_5 \sum_{i=1}^{n} (y_i - \widehat{y_i}) \cdot x_1$$

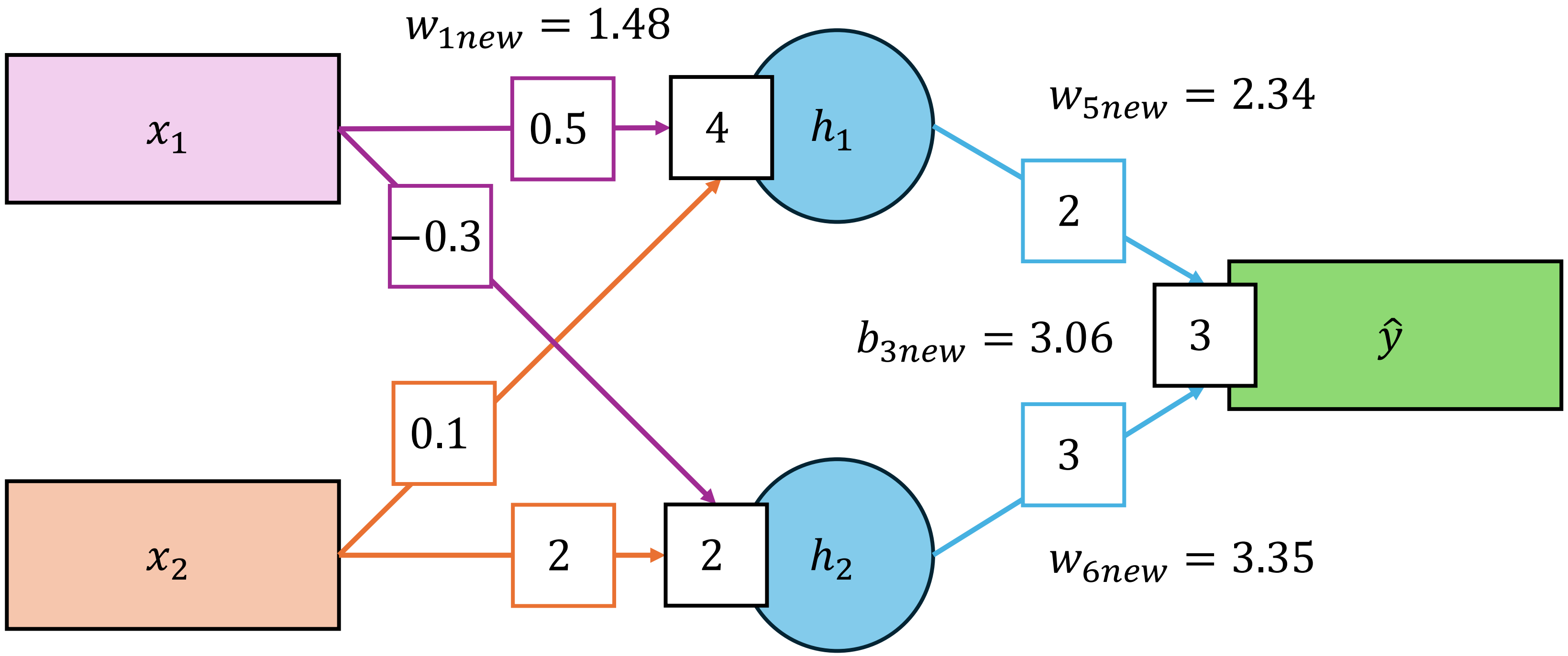
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_1} = -\frac{2}{3} (2)[(54.6)(16) + (39.1)(1) + (-11.2)(16)] = -978$$

W_1



$$w_{1new} = 0.5 - (0.001)(-978) = 1.48$$

$\boldsymbol{w_2}$



$$w_{2new} = w_{2current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_2}$$

Apply Chain Rule!!!

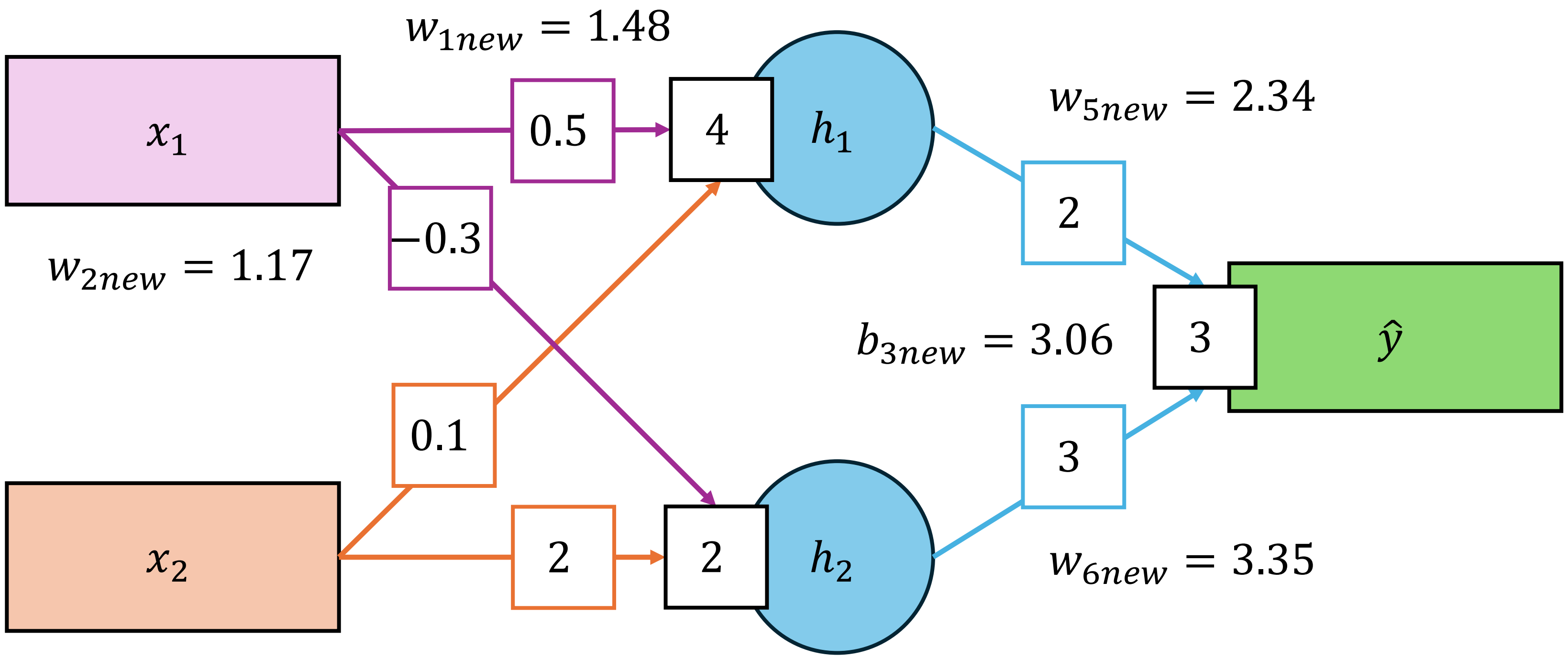
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_2} = \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial w_2}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_2} = \frac{1}{3} \sum_{i=1}^{n} \frac{\partial (y_i - \widehat{y_i})^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_5 + h_2 w_6 + b_3)}{\partial h_2} \cdot \frac{\partial (x_1 w_2 + x_2 w_4 + b_2)}{\partial w_2}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_2} = -\frac{2}{3} \cdot w_6 \sum_{i=1}^{n} (y_i - \widehat{y_i}) \cdot x_1$$

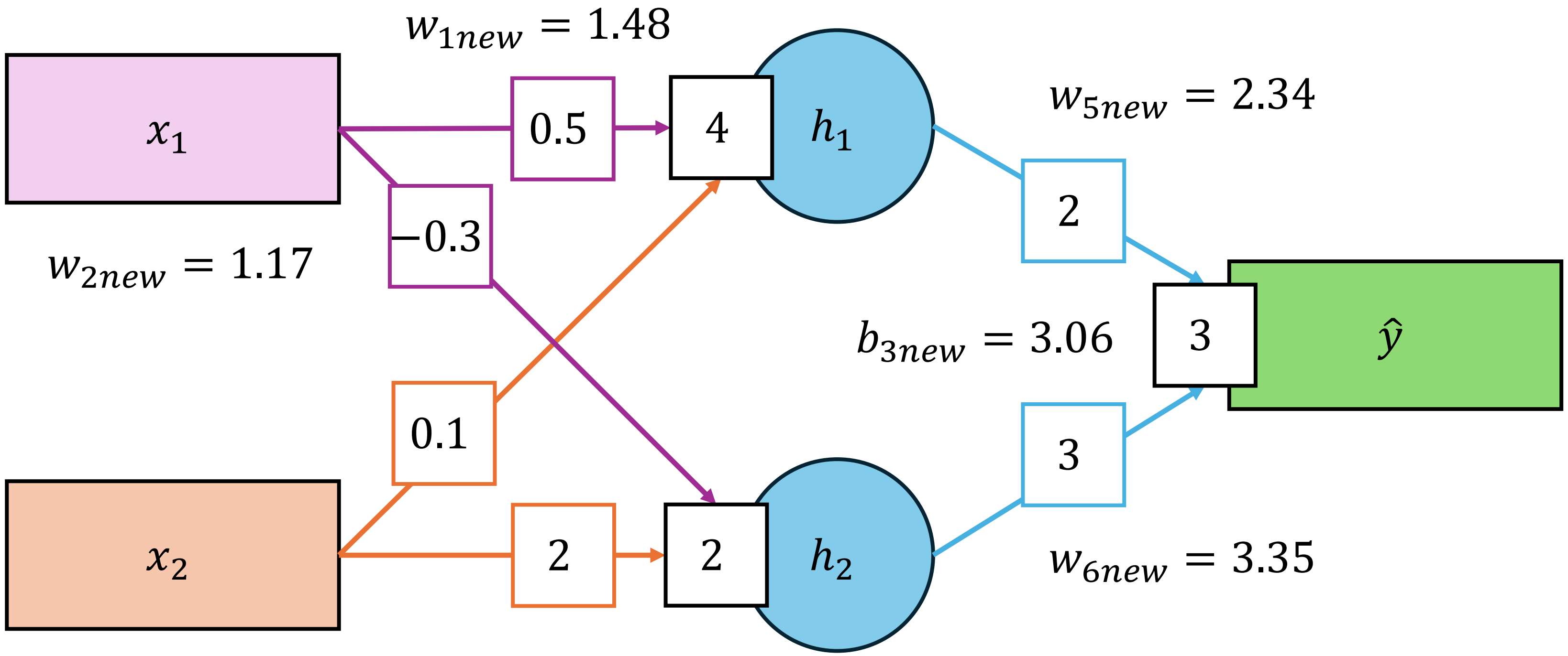
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_2} = -\frac{2}{3}(3)[(54.6)(16) + (39.1)(1) + (-11.2)(16)] = -1467$$

$\boldsymbol{W_2}$



$$w_{2new} = -0.3 - (0.001)(-1467) = 1.17$$

W_3



$$w_{3new} = w_{3current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_3}$$

Apply Chain Rule!!!

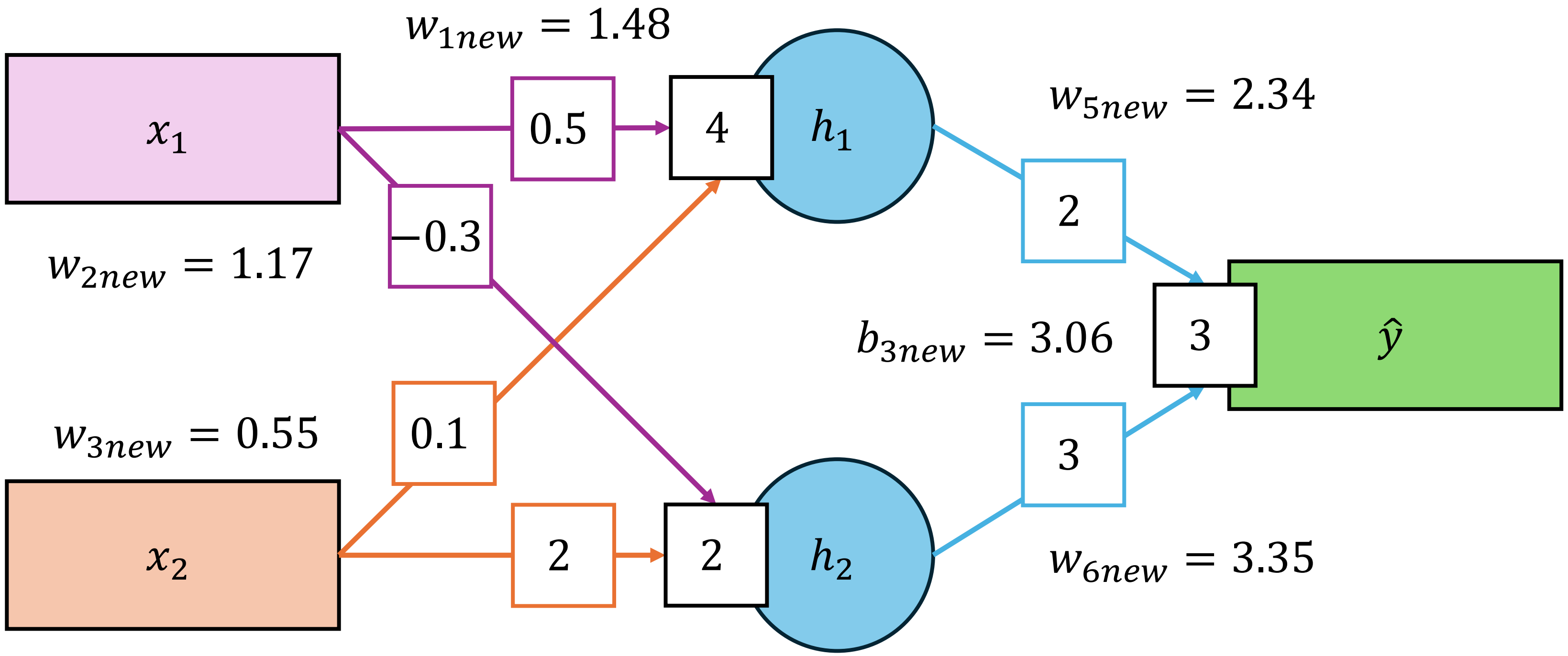
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_3} = \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_3}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_3} = \frac{1}{3}\sum_{i=1}^{n} \frac{\partial (y_i - \widehat{y_i})^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_5 + h_2 w_6 + b_3)}{\partial h_1} \cdot \frac{\partial (x_1 w_1 + x_2 w_3 + b_1)}{\partial w_3}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_3} = -\frac{2}{3} \cdot w_5 \sum_{i=1}^{n} (y_i - \widehat{y_i}) \cdot x_2$$

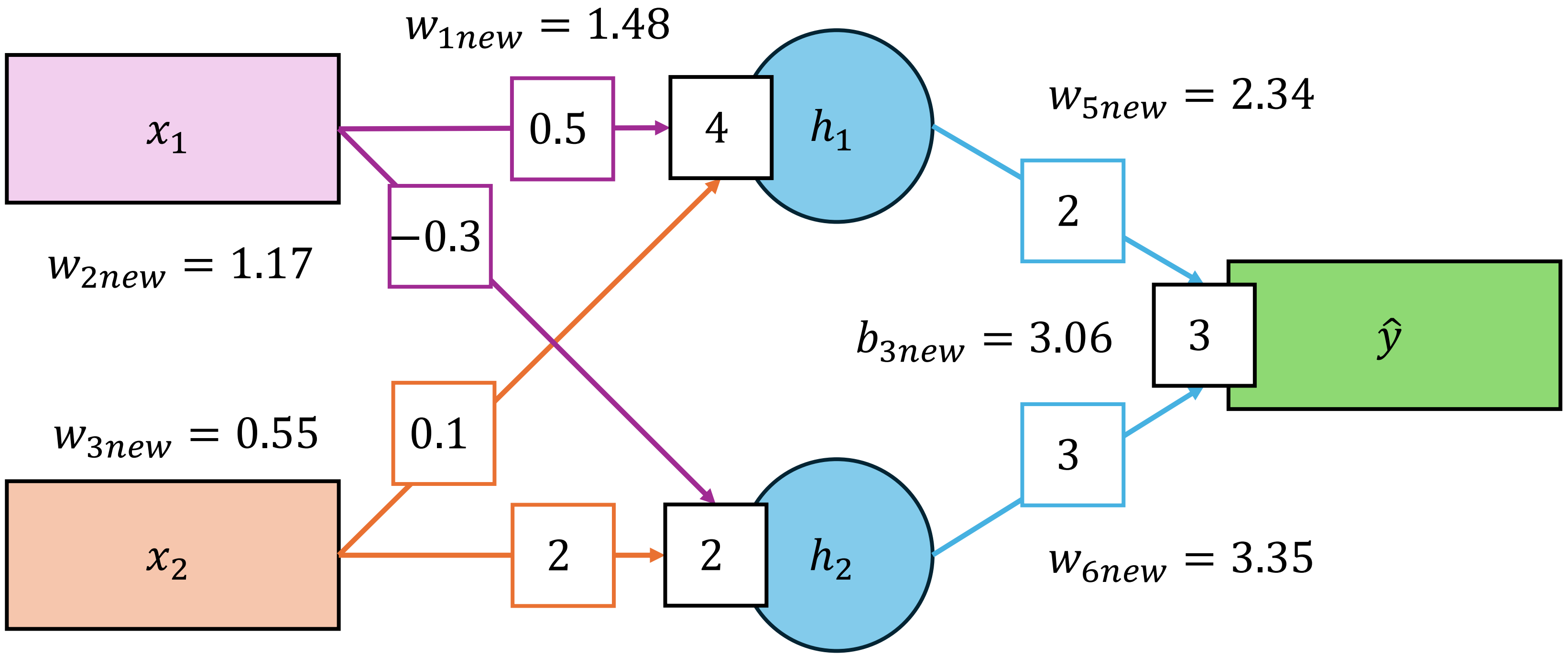
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_3} = -\frac{2}{3}(2)[(54.6)(4) + (39.1)(4) + (-11.2)(3)] = -454.93$$

W_3



$$w_{3new} = 0.1 - (0.001)(-454.93) = 0.55$$

w_4



$$w_{4new} = w_{4current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_4}$$

Apply Chain Rule!!!

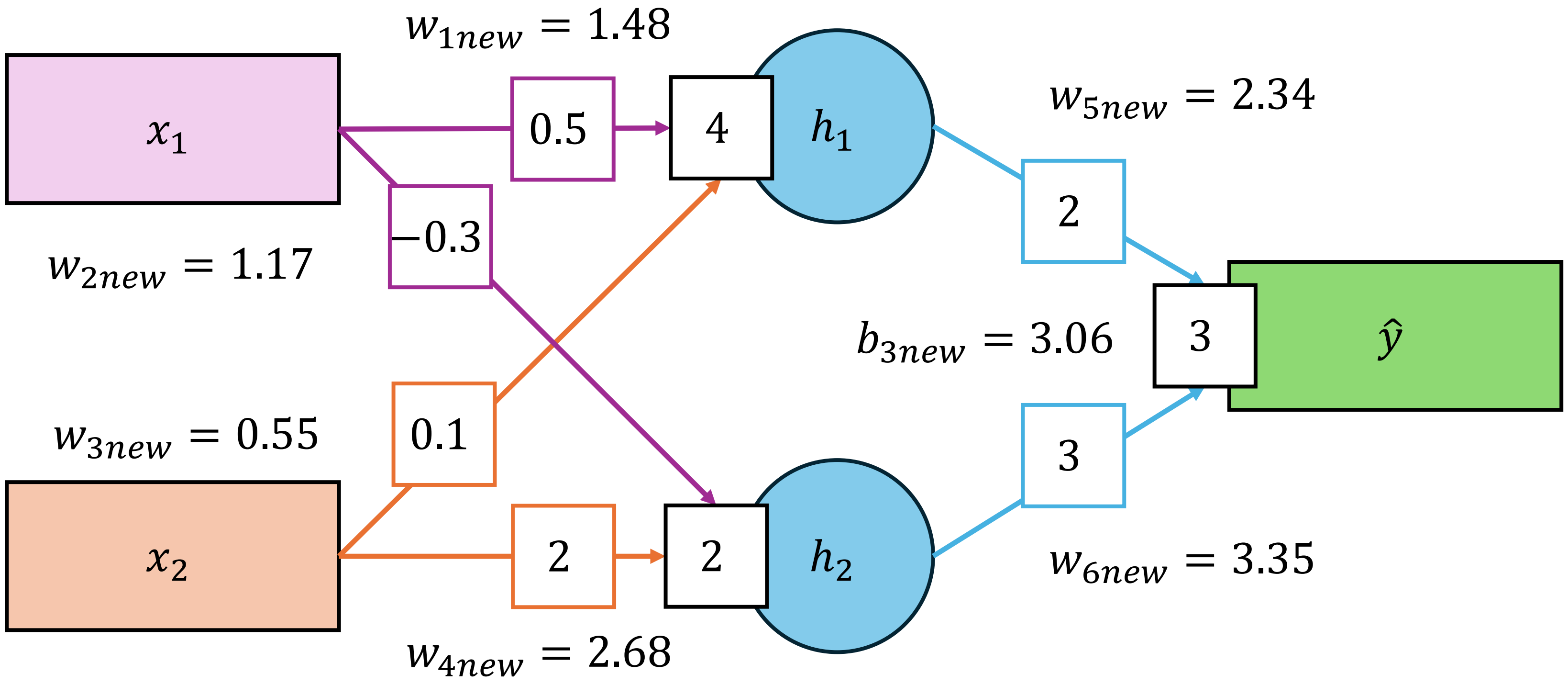
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_4} = \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial w_4}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_4} = \frac{1}{3} \sum_{i=1}^{n} \frac{\partial (y_i - \widehat{y_i})^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_5 + h_2 w_6 + b_3)}{\partial h_2} \cdot \frac{\partial (x_1 w_2 + x_2 w_4 + b_2)}{\partial w_4}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_4} = -\frac{2}{3} \cdot w_6 \sum_{i=1}^{n} (y_i - \widehat{y_i}) \cdot x_2$$

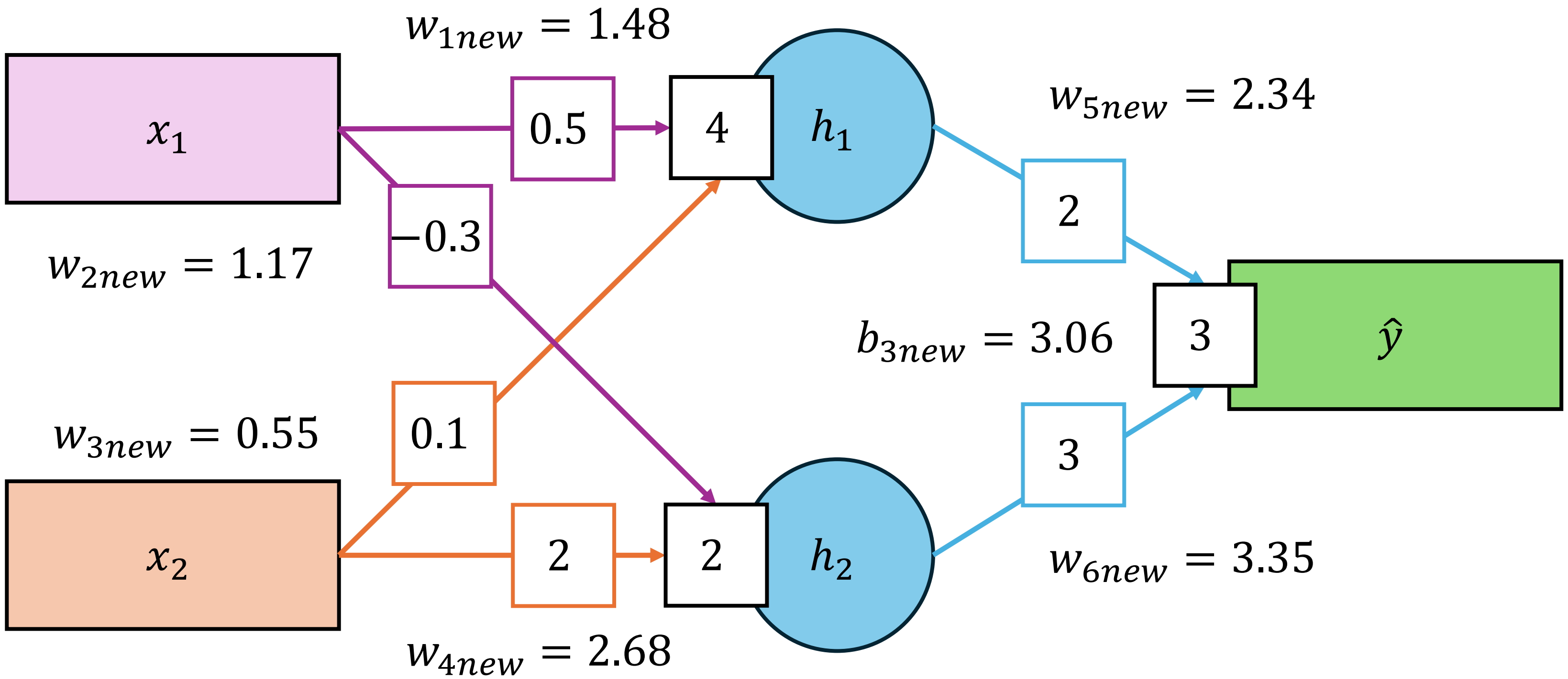
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w_4} = -\frac{2}{3}(3)[(54.6)(4) + (39.1)(4) + (-11.2)(3)] = -682.4$$

$\boldsymbol{W_4}$



$$w_{4new} = 2 - (0.001)(-682.4) = 2.68$$

b_1



$$b_{1new} = b_{1current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_1}$$

Apply Chain Rule!!!

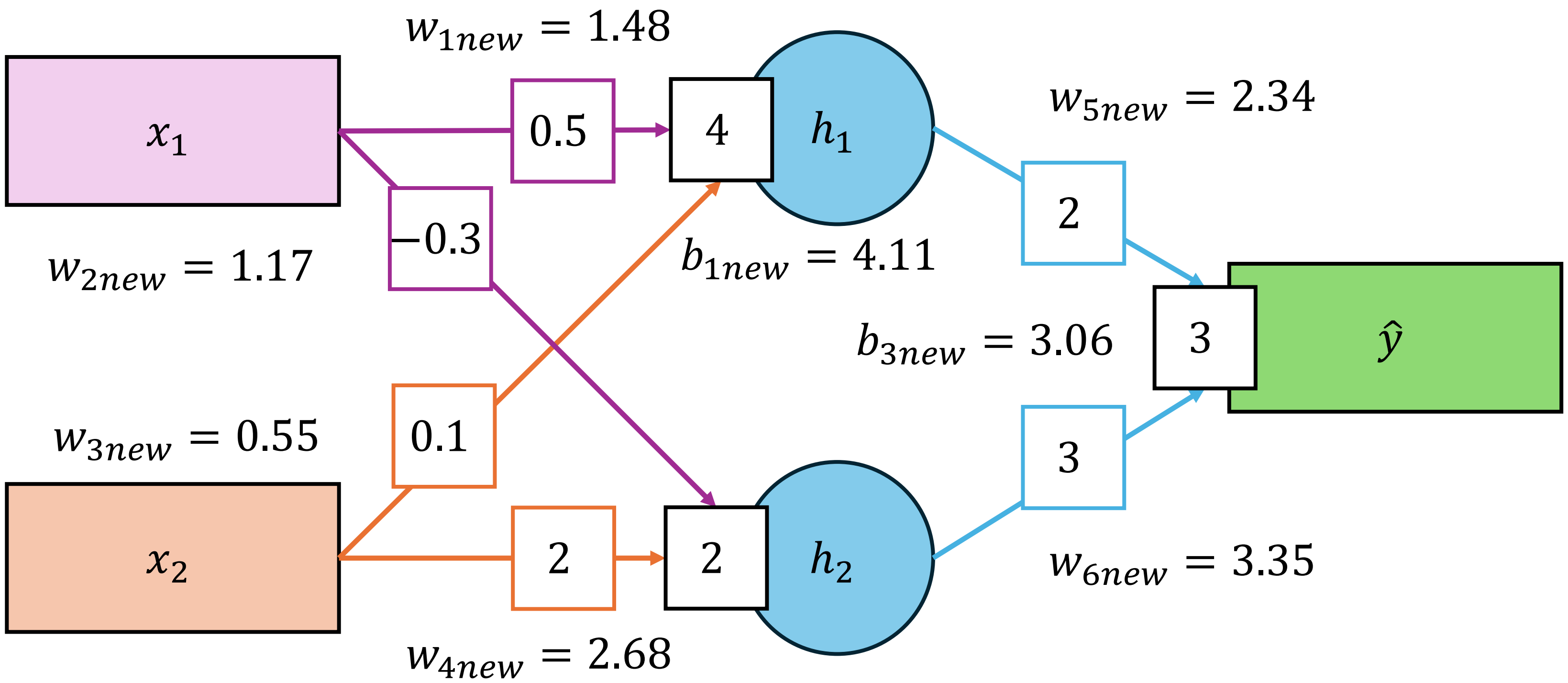
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_1} = \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_1} \cdot \frac{\partial h_1}{\partial b_1}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_1} = \frac{1}{3} \sum_{i=1}^{n} \frac{\partial (y_i - \widehat{y_i})^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_5 + h_2 w_6 + b_3)}{\partial h_1} \cdot \frac{\partial (x_1 w_1 + x_2 w_3 + b_1)}{\partial b_1}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_1} = -\frac{2}{3} \cdot w_5 \sum_{i=1}^{n} (y_i - \widehat{y_i})$$

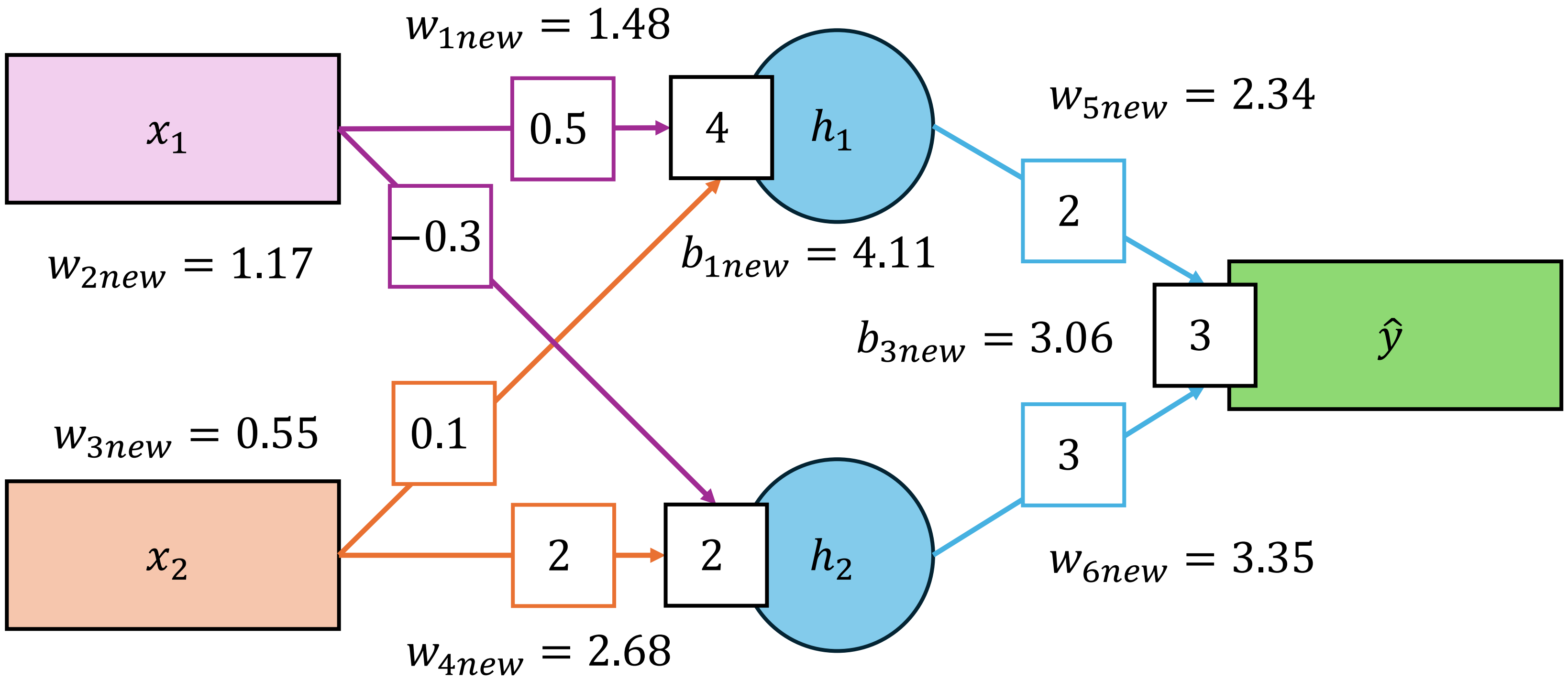
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_1} = -\frac{2}{3}(2)[(54.6) + (39.1) + (-11.2)] = -110$$

b_1



$$b_{1new} = 4 - (0.001)(-110) = 4.11$$

b_2



$$b_{2new} = b_{2current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_2}$$

Apply Chain Rule!!!

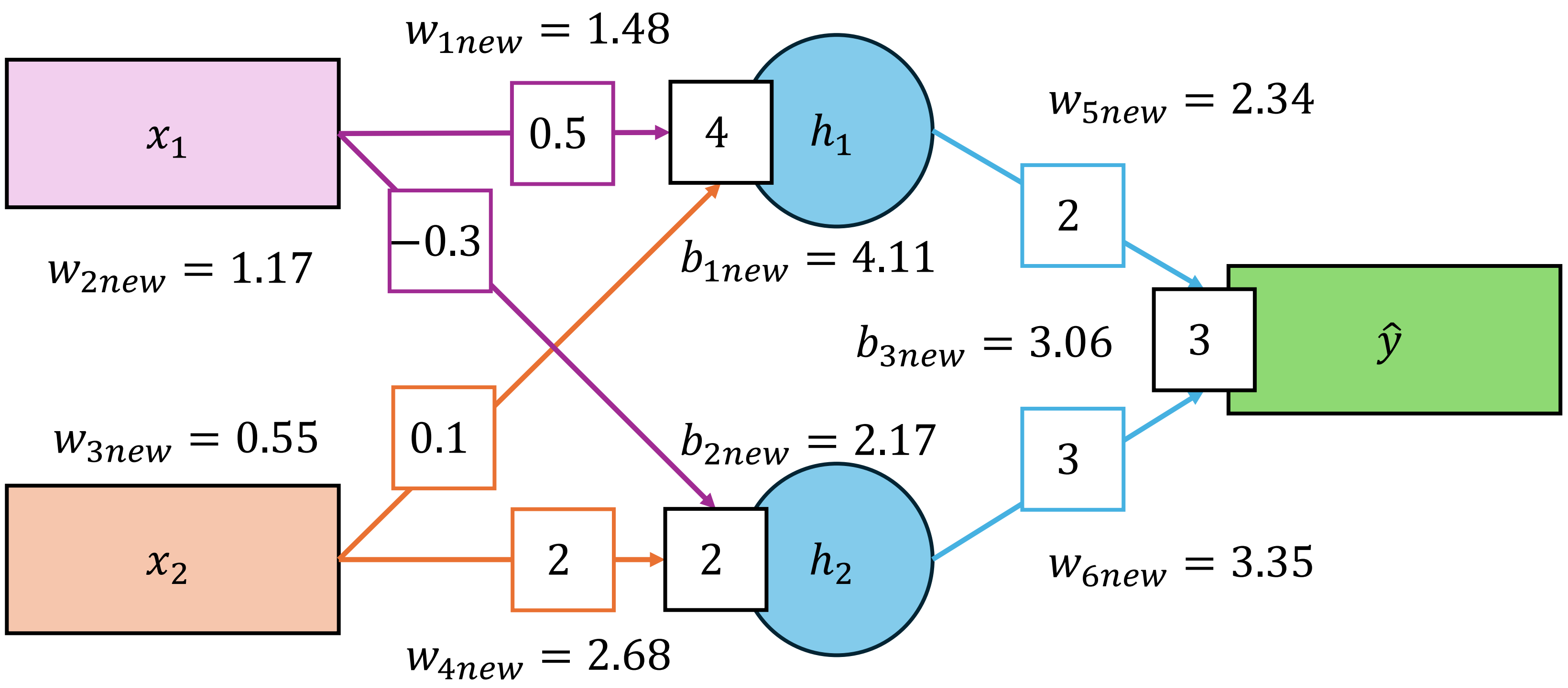
$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_2} = \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial b_2}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_2} = \frac{1}{3} \sum_{i=1}^{n} \frac{\partial (y_i - \widehat{y_i})^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_5 + h_2 w_6 + b_3)}{\partial h_2} \cdot \frac{\partial (x_1 w_2 + x_2 w_4 + b_2)}{\partial b_2}$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_2} = -\frac{2}{3} \cdot w_6 \sum_{i=1}^{n} (y_i - \widehat{y_i})$$

$$\sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b_2} = -\frac{2}{3}(3)[(54.6) + (39.1) + (-11.2)] = -165$$

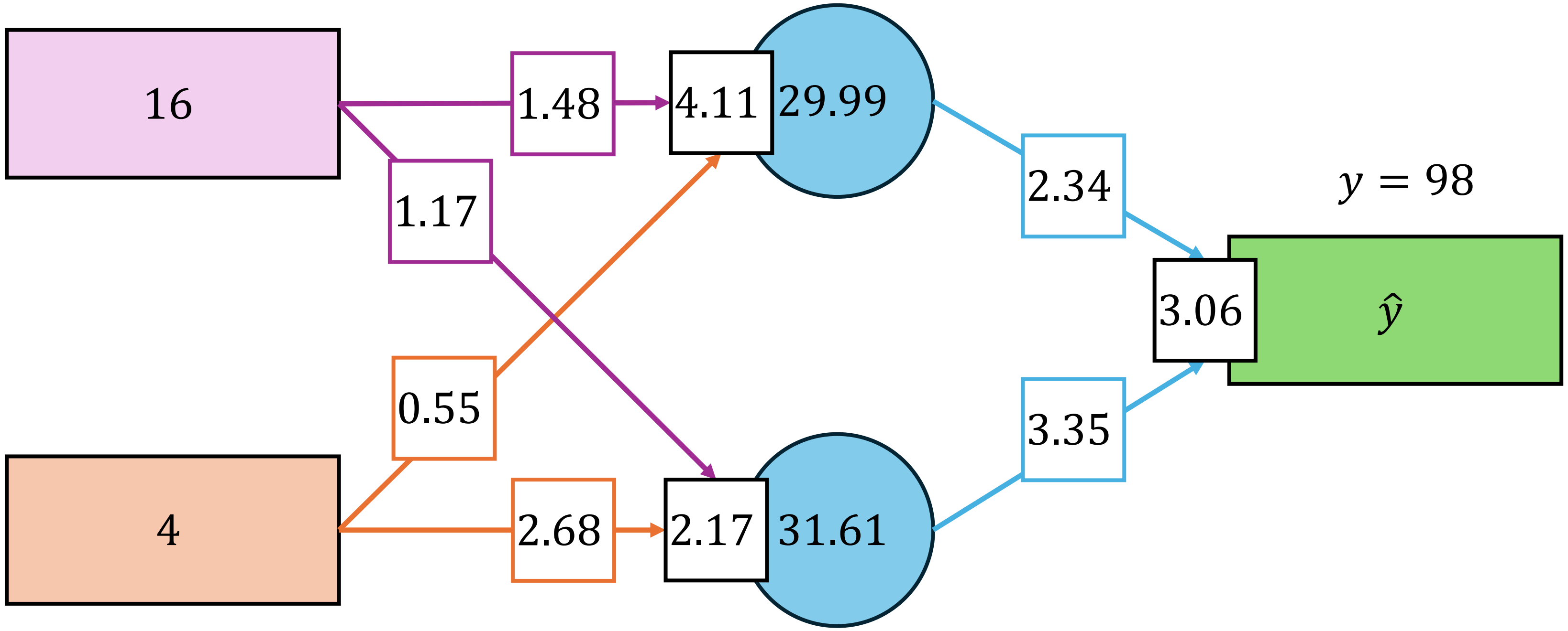
$\boldsymbol{b_2}$



$$b_{2new} = 2 - (0.001)(-165) = 2.17$$

End of Epoch: 1

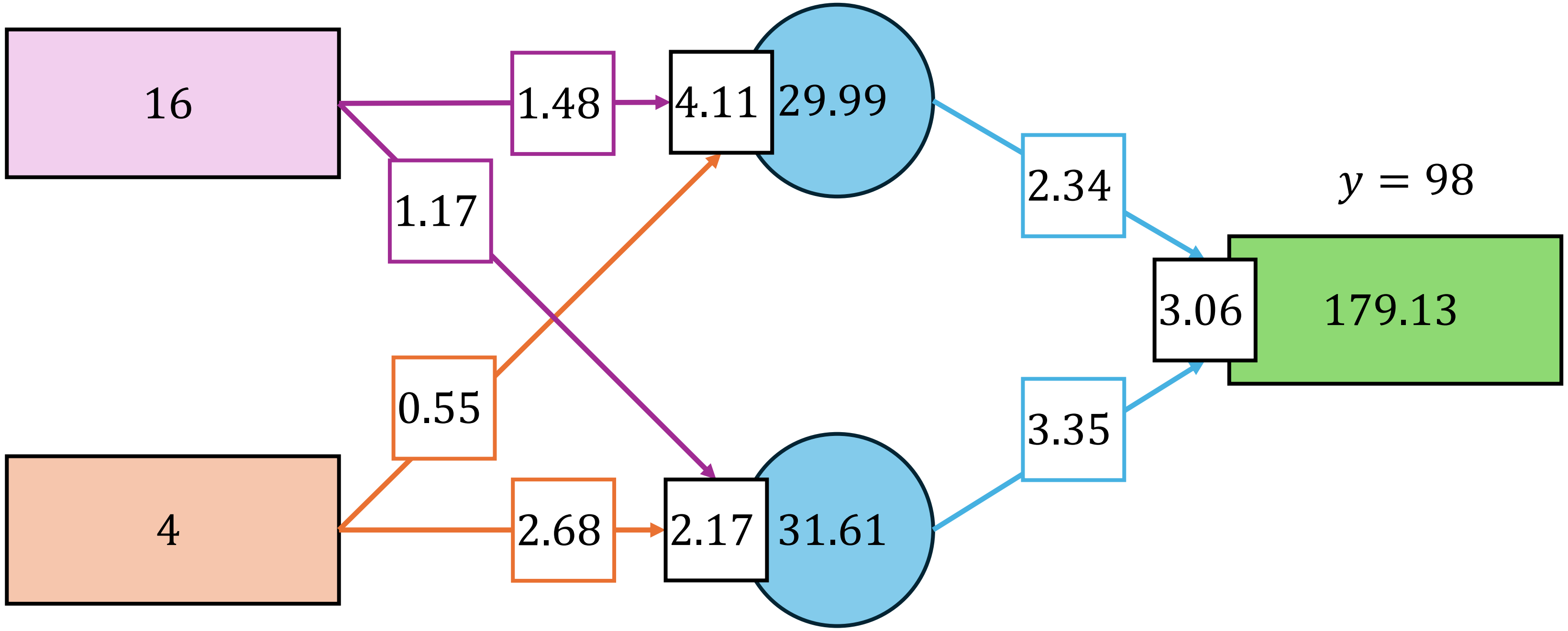
Iteration: 1, **Epoch**: 2



$$[h_1 \quad h_2] = [x_1 \quad x_2] \times \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} + [b_1 \quad b_2]$$

$$[h_1 \quad h_2] = [16 \quad 4] \times \begin{bmatrix} 1.48 & 1.17 \\ 0.55 & 2.68 \end{bmatrix} + [4.11 \quad 2.17] = [29.99 \quad 31.61]$$

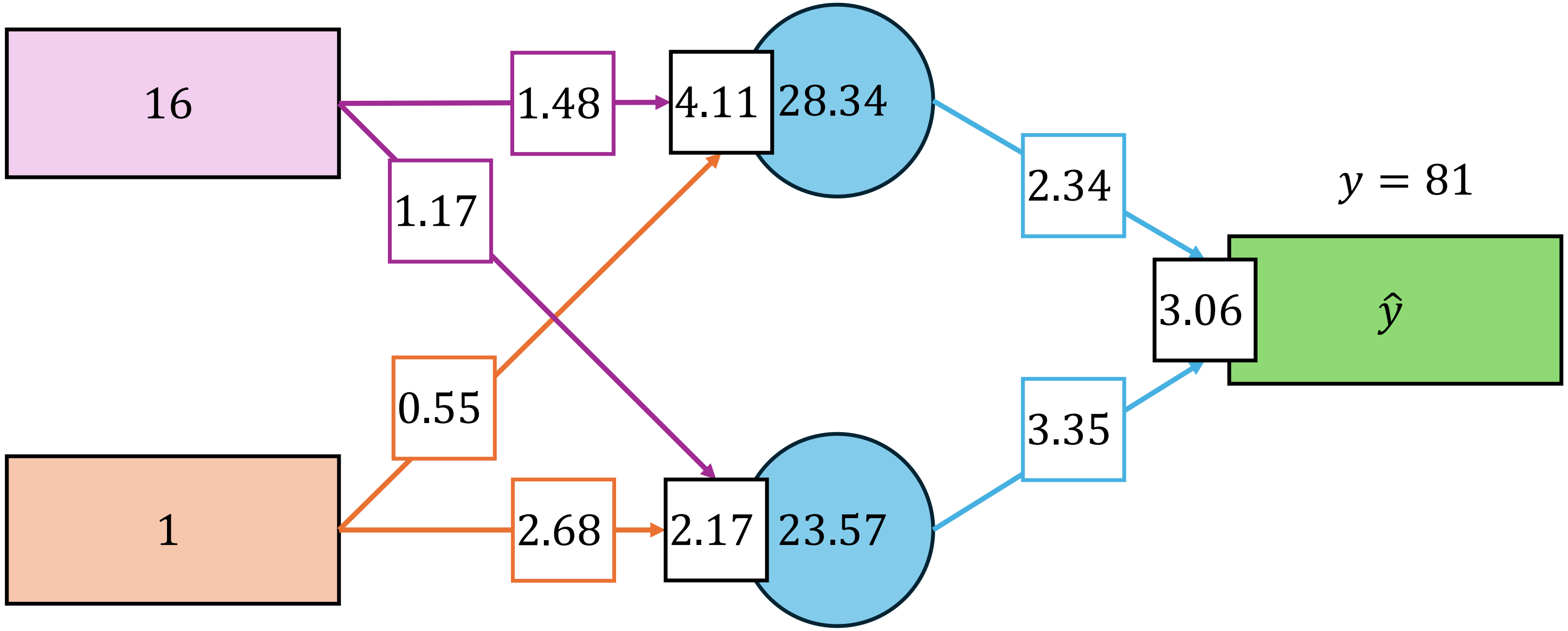
Iteration: 1, **Epoch**: 2



$$[\hat{y}] = [h_1 \quad h_2] \times \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} + [b_3]$$

$$[\hat{y}] = [29.99 \quad 31.61] \times \begin{bmatrix} 2.34 \\ 3.35 \end{bmatrix} + [3.06] = [179.13]$$

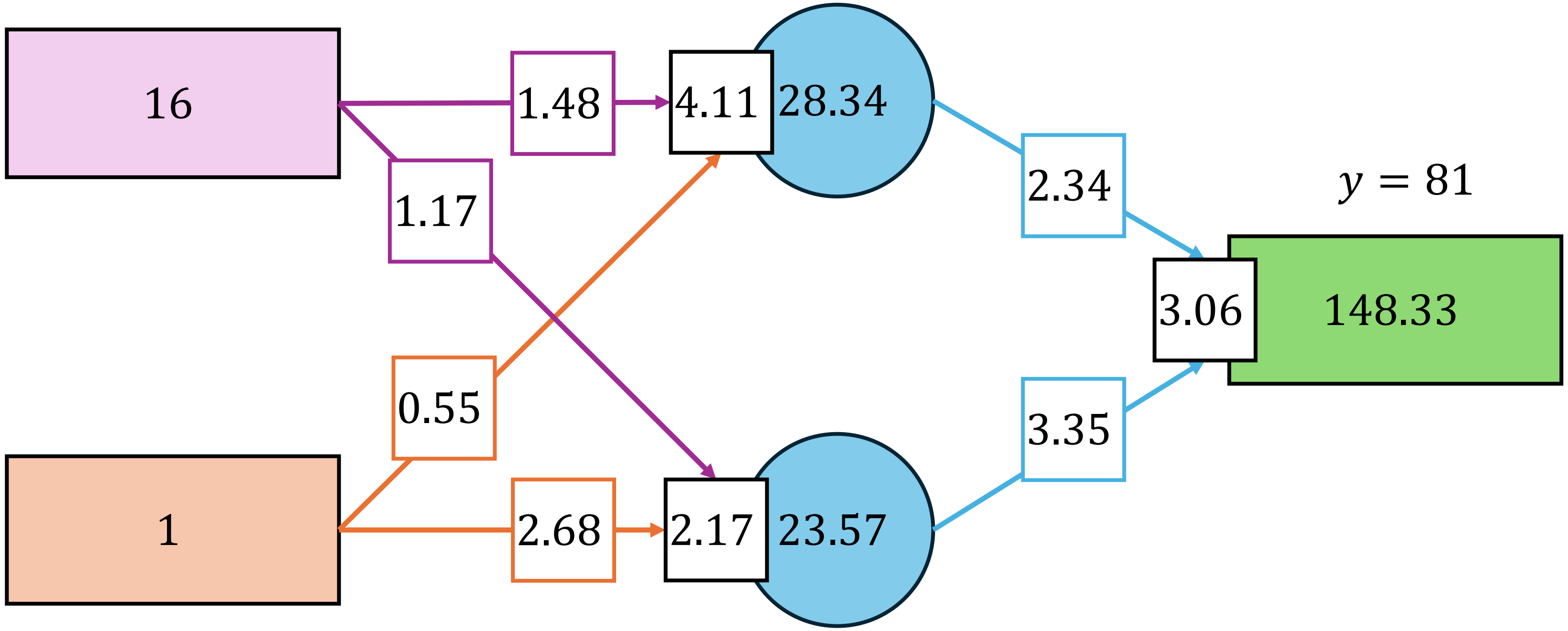
Iteration: 2, **Epoch**: 2



$$[h_1 \quad h_2] = [x_1 \quad x_2] \times \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} + [b_1 \quad b_2]$$

$$[h_1 \quad h_2] = [16 \quad 1] \times \begin{bmatrix} 1.48 & 1.17 \\ 0.55 & 2.68 \end{bmatrix} + [4.11 \quad 2.17] = [28.34 \quad 23.57]$$

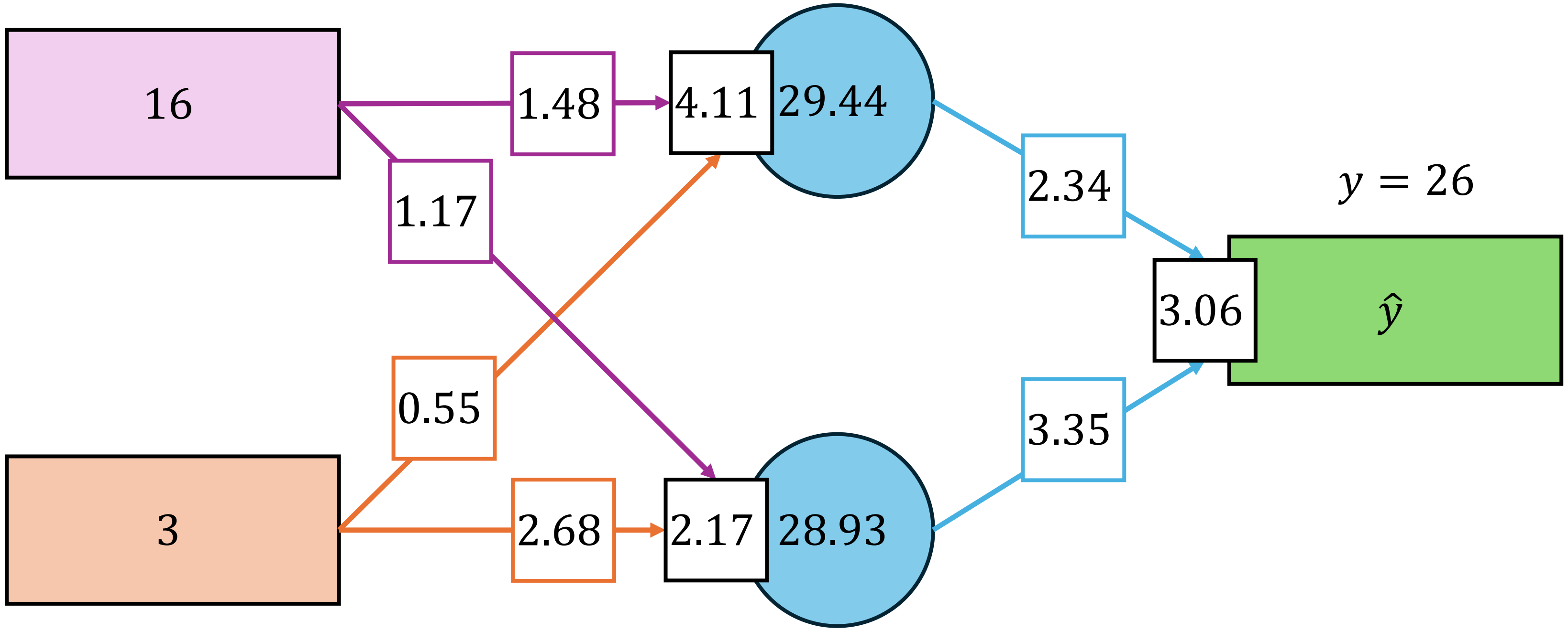
Iteration: 2, **Epoch**: 2



$$[\hat{y}] = [h_1 \quad h_2] \times \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} + [b_3]$$

$$[\hat{y}] = [28.34 \quad 23.57] \times \begin{bmatrix} 2.34 \\ 3.35 \end{bmatrix} + [3.06] = [148.33]$$

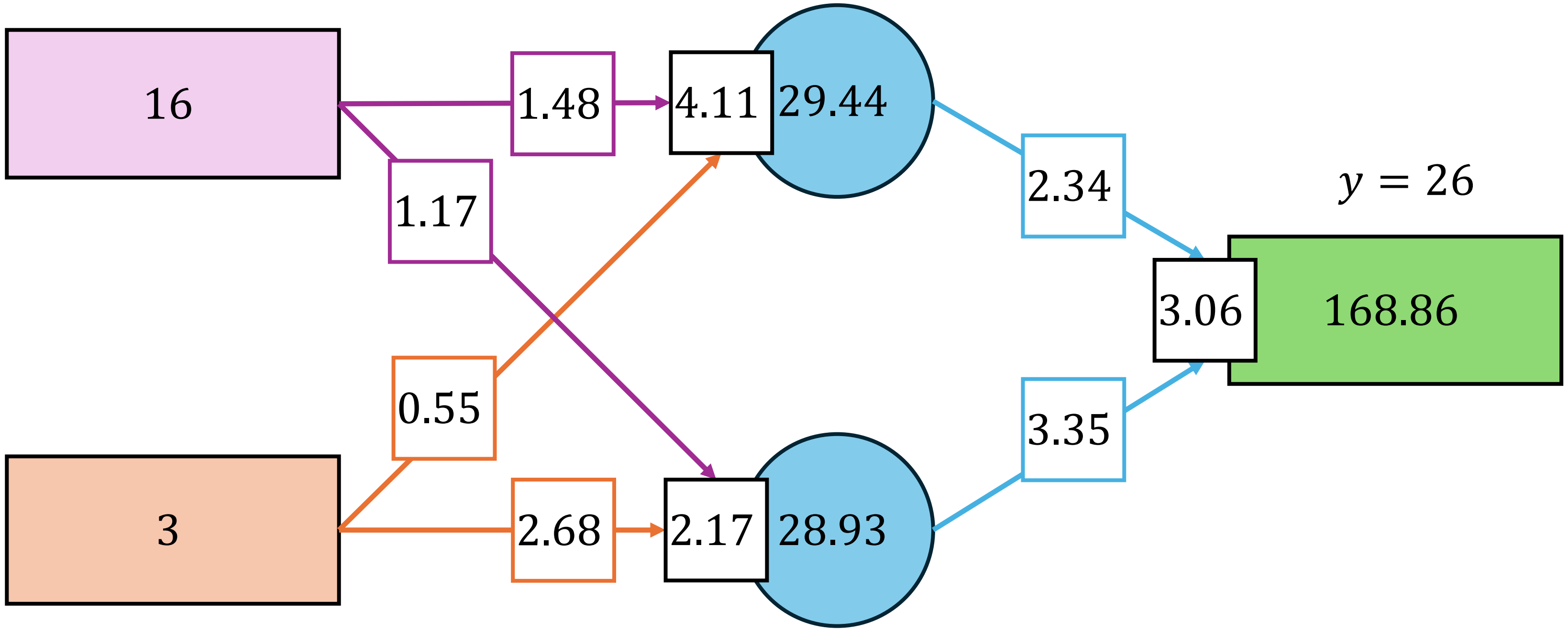
Iteration: 3, **Epoch**: 2



$$[h_1 \quad h_2] = [x_1 \quad x_2] \times \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} + [b_1 \quad b_2]$$

$$[h_1 \quad h_2] = [16 \quad 3] \times \begin{bmatrix} 1.48 & 1.17 \\ 0.55 & 2.68 \end{bmatrix} + [4.11 \quad 2.17] = [29.44 \quad 28.93]$$

Iteration: 3, **Epoch**: 2



$$[\hat{y}] = [h_1 \quad h_2] \times \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} + [b_3]$$

$$[\hat{y}] = [29.44 \quad 28.93] \times \begin{bmatrix} 2.34 \\ 3.35 \end{bmatrix} + [3.06] = [168.86]$$

i	x_1	x_2	y	h_1	h_2	$\hat{y}$	$(y_i - \hat{y}_i)$
1	16	4	98	39.99	31.61	179.13	-81.13
2	1	4	81	4.9	28.34	148.33	-67.33
3	16	3	26	29.44	28.93	168.86	-142.86

Total Loss

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$MSE = \frac{1}{3}\left((-81.13\,)^2 + (-67.33\,)^2 + (-142.86\,)^2\right) = 10508.13$$

Ahhhhh!!!

Time to code in

PyTorch

1. Import Library

```
import torch
import torch.nn as nn
import torch.optim as optim
import torch.nn.functional as F
import numpy as np
```

[1]

Import library

```
device = ("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using {device} device")
```

[2]

```
Using cuda device
```

Declare to using **GPU,** in case GPU nor found it's going to use CPU instead.

2. Define Train|Test Data

```python
training_x = torch.tensor([
    [16., 4.],
    [1.,  4.],
    [16., 3.]
])

training_y = torch.tensor([
    [98.],
    [81.],
    [26.]
])
```

[3] ✓ 0.0s

Training Data

```python
testing_x = torch.tensor([
    [1., 3.]
])

testing_y = torch.tensor([
    [3.]
])
```

[4] ✓ 0.0s

Testing Data

2. Define Train|Test Data

```
dataset = TensorDataset(training_x, training_y)
train_loader = DataLoader(dataset, batch_size=1, shuffle=True)
```

[5] ✓ 0.0s

Put training data into **DataLoader**

3. Define Model

```
class ANNModeler(nn.Module):
    def __init__(self, input_size, output_size):
        super(ANNModeler, self).__init__()
        self.linear1 = nn.Linear(input_size, 2, bias=True)
        self.linear2 = nn.Linear(2, output_size, bias=True)

    def forward(self, x):
        out = self.linear1(x)
        return self.linear2(out)
```

[6] ✓ 0.0s

$$[h_1 \quad h_2] = [x_1 \quad x_2] \times \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix} + [b_1 \quad b_2]$$

$$[\hat{y}] = [h_1 \quad h_2] \times \begin{bmatrix} w_5 \\ w_6 \end{bmatrix} + [b_3]$$

Forward propagation

4. Setup Loss Function and Optimizer

```python
losses = []
model = ANNModeler(2, 1).to(device)
loss_function = nn.MSELoss()
optimizer = optim.SGD(model.parameters(), lr=0.000001)

print(model)
```

[7] ✓ 0.7s

```
ANNModeler(
  (linear1): Linear(in_features=2, out_features=2, bias=True)
  (linear2): Linear(in_features=2, out_features=1, bias=True)
)
```

Train model using GPU

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$w_{new} = w_{current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial w}$$

$$b_{new} = b_{current} - \alpha \sum_{i=1}^{n} \frac{\partial MSE_i}{\partial b}$$

5. Training Step

```python
epochs = 1000

for epoch in range(epochs):
    total_loss = 0

    for (x, y) in train_loader:
        x, y = torch.tensor(x).to(device), torch.tensor(y).to(device)

        model.zero_grad()

        y_hat = model(x)

        loss = loss_function(y, y_hat)

        loss.backward()
        optimizer.step()
        total_loss += loss.item()

    losses.append(total_loss)
```

[8] ✓ 2.7s

Training data

Zero gradient check

Comparing losses

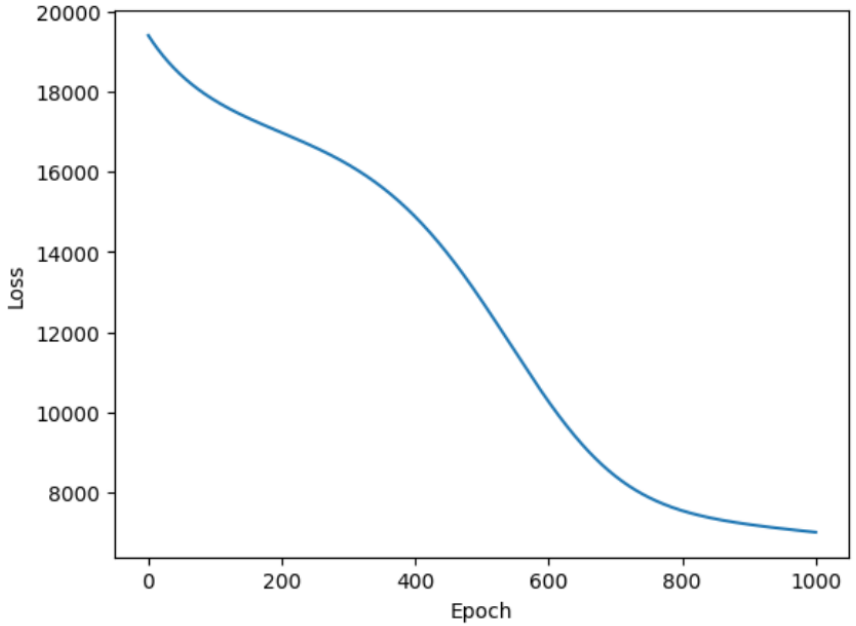
Backpropagation

6. Losses Plotting

```python
import matplotlib.pyplot as plt
```
[9] ✓ 0.3s

```python
def plot_losses(ax, t, losses):
    ax.plot(t, losses)
    ax.set_xlabel("Epoch")
    ax.set_ylabel("Loss")
```
[10] ✓ 0.0s

```python
fig, ax = plt.subplots()
plot_losses(ax, np.linspace(0., len(losses), len(losses)), losses)
```
[11] ✓ 0.1s



7. Result Inspection

```
for x, y in zip(testing_x, testing_y):
    # Get predicted vector
    pred = model(torch.tensor(x).to(device))

    print(f"y_true: {int(y.item())}, y_hat: {int(pred.item())}")
```

[12] ✓ 0.0s

```
y_true: 3, y_hat: 11
C:\Users\HashTable\AppData\Local\Temp\ipykernel_18900\76514697.py:3: Use
  pred = model(torch.tensor(x).to(device))
```