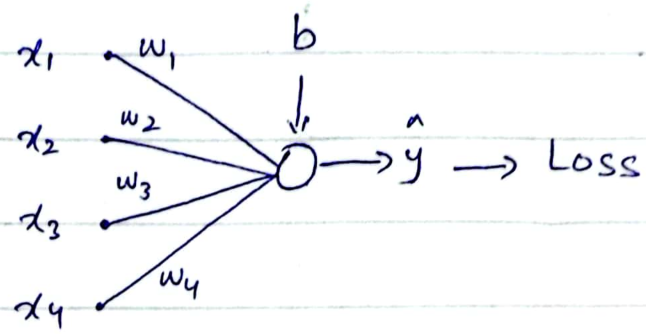
Single Neuron Neural Network

This is a simple neural network with a single neuron and no activation function.



For this exercise, we assume four inputs, so we have four weights and one bias.

Data

$$\underbrace{\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix}}_{X} \qquad \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}}_{Y}$$

Shape of dataset (Features) : 3×4

It means ~~we~~ features matrix has 3 rows and 4 columns.

Since, there are four features, so the no. of weights will be 4 and one bias:

Weight matrix

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

bias

(b).

Shape of weight matrix : 4 x 1.

Now, we are ready for forward propagation (forward pass).

Forward Propagation formula

$$\hat{y} = w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + w_4 x_{i4} + b$$

Since our feature matrix contains more than 1 row, we take assistance from Linear Algebra.

Performing matrix multiplication

$$\hat{y} = \left(\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix}_{3\times 4} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}_{4\times 1} \right) + b$$

This becomes...

$$\begin{bmatrix} x_{11}w_1 + x_{12}w_2 + x_{13}w_3 + x_{14}w_4 + b \\ x_{21}w_1 + x_{22}w_2 + x_{23}w_3 + x_{24}w_4 + b \\ x_{31}w_1 + x_{32}w_2 + x_{33}w_3 + x_{34}w_4 + b \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \end{bmatrix}$$

Forward Propagation is done. It is time to calculate loss. Our priority is given to mean squared error (MSE)

MSE Formula

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

where N = No. of rows.

So, the loss will be

$$\text{Loss}(\text{mse}) = \frac{1}{3}\left[(y_1-\hat{y}_1)^2+(y_2-\hat{y}_2)+(y_3-\hat{y}_3)\right]$$

The result of this formula will be a single number.

Now, it is turn for back-propagation.

$$\boxed{\text{Please refer some article/YT video to learn back-propagation}}$$

Perform partial differentiation of Loss with respect to weights. I am going to do it for one weight only.

$$\boxed{\begin{array}{l} \text{Loss} \rightarrow \hat{y} \rightarrow w_1 \\ \qquad\qquad \hat{y} \rightarrow x_1 \\ \qquad\qquad \hat{y} \rightarrow b \end{array}}$$

$\leftarrow$ Dependency graph

$\uparrow$

I gave this name :)

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_1}$$

$$\frac{\partial L}{\partial \hat{y}} = \frac{\partial((y_1-\hat{y}_1)^2)}{\partial \hat{y}} = 2(y_1-\hat{y}_1)(-1) \Rightarrow -2(y_1-\hat{y}_1)$$

$$\frac{\partial \hat{y}}{\partial w_1} = \cancel{\frac{\partial L}{\partial \hat{y}}} = \frac{\partial(w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + w_4 x_{i4} + b)}{\partial w_1}$$

$$\frac{\partial \hat{y}}{\partial w_1} = x_{i1}$$

$$\boxed{\frac{\partial L}{\partial w_1} = -2(y_1-\hat{y}_1) \cdot x_{i1}}$$

$$\boxed{\frac{\partial L}{\partial b} = -2(y_1-\hat{y}_1)}$$

However, for our dataset, it is...

$$\frac{\partial L}{\partial w} = \frac{-2}{3}(y-\hat{y})\, x$$

$$\frac{\partial L}{\partial b} = \frac{-2}{3} \cdot \sum_{i=1}^{N} (y_i - \hat{y}_i)$$

Let's see how it works...

$$\frac{\partial L}{\partial \hat{y}} = \frac{-2}{3} \cdot \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ y_3 - \hat{y}_3 \end{bmatrix}_{3\times 1} \Rightarrow \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}_{3\times 1}$$

$$\frac{\partial L}{\partial w} = x^T @ \frac{\partial L}{\partial \hat{y}}$$ $\because$ @ = Dot Product

$$= \begin{bmatrix} x_{11} & x_{21} & x_{31} \\ x_{12} & x_{22} & x_{32} \\ x_{13} & x_{23} & x_{33} \\ x_{14} & x_{24} & x_{34} \end{bmatrix}_{4\times 3} @ \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}_{3\times 1}$$

$$= \begin{bmatrix} x_{11}c_1 + x_{12}c_2 + x_{31}c_3 \\ x_{12}c_1 + x_{22}c_2 + x_{32}c_3 \\ x_{13}c_1 + x_{23}c_2 + x_{33}c_3 \\ x_{14}c_1 + x_{24}c_2 + x_{34}c_4 \end{bmatrix}_{4\times 1} = \begin{bmatrix} w_1' \\ w_2' \\ w_3' \\ w_4' \end{bmatrix}_{4\times 1}$$

I should have mentioned it earlier...

Updation Formula

$$w_n = w_o - \eta \frac{\partial L}{\partial w}$$ ∵ w_n = new weights, w_o = old weights

η = learning rate

$$b_n = b_o - \eta \frac{\partial L}{\partial b}$$ ∵ b_n = new bias, b_o = old bias

So, in our example

$$\begin{bmatrix} w_{1n} \\ w_{2n} \\ w_{3n} \\ w_{4n} \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} - \eta \begin{bmatrix} w_1' \\ w_2' \\ w_3' \\ w_4' \end{bmatrix}$$

All weights are updated.

$$\frac{\partial L}{\partial b} = \frac{-2}{3}\left[(y_1-\hat{y}_1)+(y_2-\hat{y}_2)+(y_3-\hat{y}_3)+ y_4-\hat{y}_4)\right]$$

$$= b'$$

$b_n = b_0 - \eta b'$

Finally, bias is also updated.

Remember :- Weights and biases are also known as trainable parameters.

In the next iteration,

$\hat{y} = x \cdot w_n + b_n$

This process will continue until we reach a specific no. of iterations or a loss value.