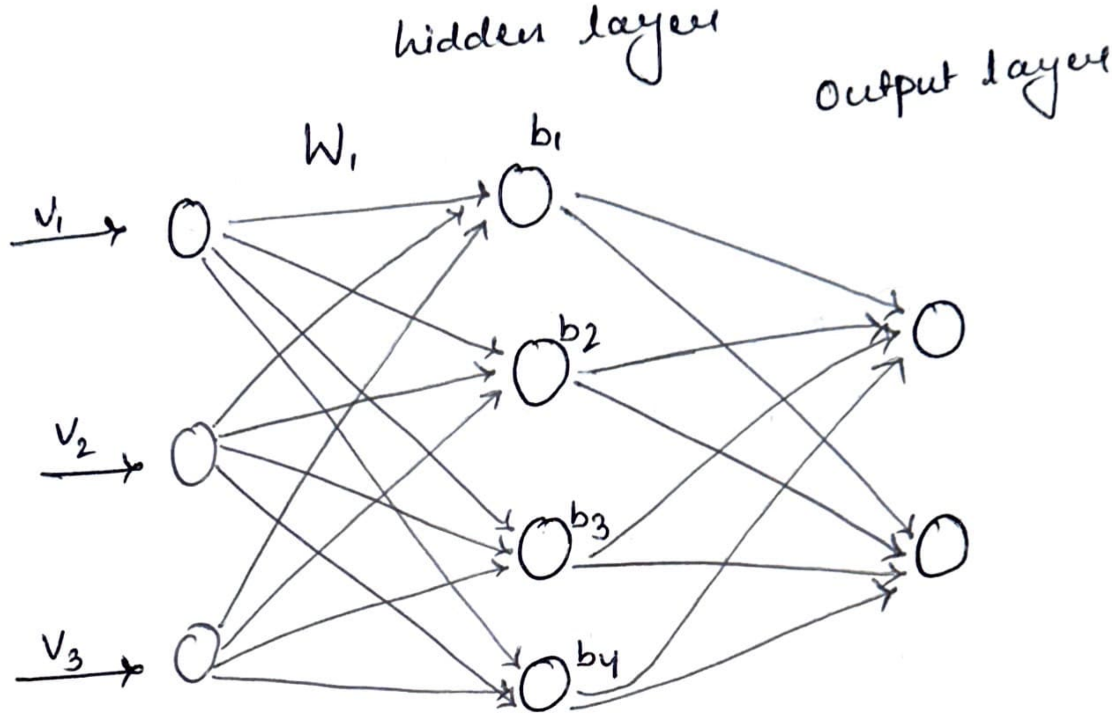
For a 2-layer neural Network, with 1 hidden layer and 1 output layer.

Fig:- Neural Network :—



Forward Propagation :—

The values for each output neuron can be calculated as $y_i = \sum_i x_i w_{ij} + b_j$

Dot product for forward propagation can be calculated as

$$X = [x_1 \ldots x_i] \quad W = \begin{bmatrix} w_{11} & \cdots & w_{1j} \\ \vdots & & \\ w_{i1} & \cdots & w_{ij} \end{bmatrix}$$

$$B = [b_1 \ldots b_j]$$

Gradient Descent

$$w \leftarrow w \leftarrow \alpha \frac{\partial e}{\partial w}$$

Back propagation derivation :-

We need to give derivative of its error of the output which is $\partial E / \partial y$ and the back-propagation algorithm will give derivative of error of the input $\partial E / \partial x$

$$\frac{\partial E}{\partial x} \longleftarrow \text{layer} \leftarrow \frac{\partial E}{\partial y}$$

Where

$$\frac{\partial E}{\partial x} = \left[\frac{\partial E}{\partial x_1} \quad \frac{\partial E}{\partial x_2} \quad \cdots\cdots \quad \frac{\partial E}{\partial x_i}\right]$$

$$\frac{\partial E}{\partial y} = \left[\frac{\partial E}{\partial y_1} \quad \frac{\partial E}{\partial y_2} \quad \cdots\cdots \quad \frac{\partial E}{\partial y_i}\right] \text{ and}$$

E is scalar of x & y matrices.

Therefore to calculate $\partial E/\partial x$ we need to update the weights by using update ~~scale~~ rule

$$\frac{\partial E}{\partial w} = \sum_j \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial w}$$

Projection of Backpropagation

$$\frac{\partial E}{\partial x} \leftarrow \frac{\partial H_1}{\partial x} \times \frac{\partial E}{\partial H_1} \leftarrow \frac{\partial H_2}{\partial H_1} \times \frac{\partial E}{\partial H_2}$$

Now, it is needed to add activation function for every derivative of back propagation

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial y} f'(x)$$ where $f'(x)$ is

derived from every neuron

$$Y = [f(x_1) \; f(x_2) \; \ldots \ldots \; f(x_i)]$$
$$= f(x)$$

After updating weight for $\frac{dE}{dx}$ it is needed to calculate error for each output which helps to reduce the loss function

$$E = 1/n \sum_i^n (y_i^* - y_i)^2$$

after getting the error, it is ~~need~~ required to calculate mean square error where it is the mean of square of error of actual value and predicted values.

Sigmoid activation function

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

after derivating this function

$$\sigma'(x) = (1+e^{-y})(1-1+e^{-y})$$

it can be termed as ~~$f(y) = x(1-x)$~~

$$\sigma'(x) = x(1-x)$$

If we use sigmoid. this will give the log values of the output ranging from 0 to 1.

But using update rule on binary classification using log loss, the model cannot interpret the binary output which should be either 0 or 1.