Deep Learning

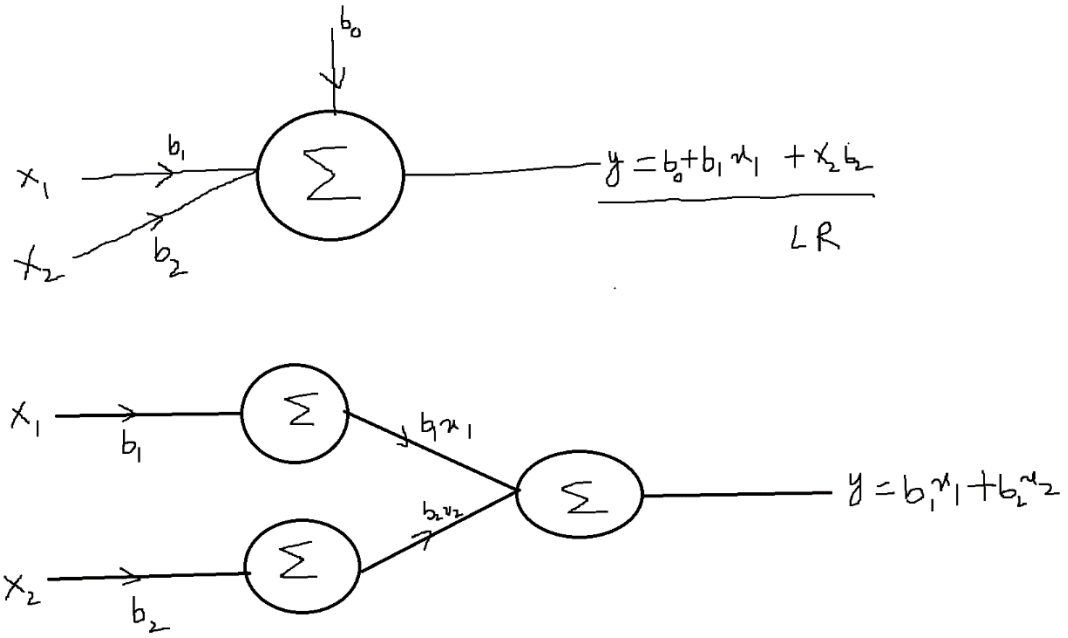
Neural Network

1) **ANN: Artificial Neural Network**
2) **DNN : Deep Neural Network**
   All types of ML model classification or Regression or Cluster
3) **CNN: Convolution Neural Network**
   Image object detection or Video annotation
   RCNN
   Fast RCNN
   MASK RNN
   Yolo
4) **RNN: Recurrent Neural Network**
   NLP process
   Next word prediction

ANN:

Neuron
Human brain with Machine Brain
Nerves axon synapse
How we are analyse a problem, what are the pattern we are generating for a solution

1) **Linear regression**
2) **Logistic regression**



Linear regression equation:

$$y = b_o + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n$$

Linear combination,

If you see above neuron diagram it is just pass the information

understand the patterns: *non linearity*

Logistic regression:

$$p(y) = sigmoid(Linear\ regression)$$

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

$$p(y) = \sigma(b_o + b_1 * x_1)$$

$$p(y) = \frac{1}{1+e^{-(bo+b_1*x_1)}}$$

$$y = b_o + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n$$

$$b_o = w_o$$

$$b_1 = w_1,\ b_2 = w_2, \ldots\ b_n = w_n$$

$$w_1, w_2, w_3 \ldots\ w_n = weights$$

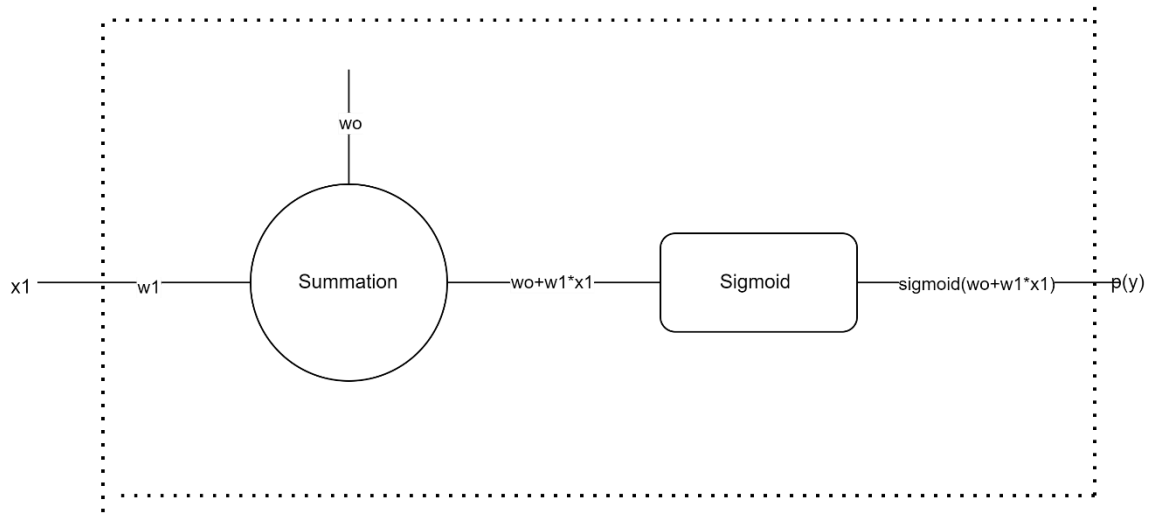
$step - 1$: $take\ the\ input\ x_1$

$step - 2$: $take\ the\ weight\ w_1$

$step - 3$: $take\ the\ bias\ w_o$

$step - 4$: $apply\ the\ Linear\ combination = w_o + w_1 * x_1$

$step - 5$: $apply\ the\ sigmoid\ on\ step - 4 = \sigma(w_o + w_1 * x_1)$



Neuron= Summation + Sigmoid

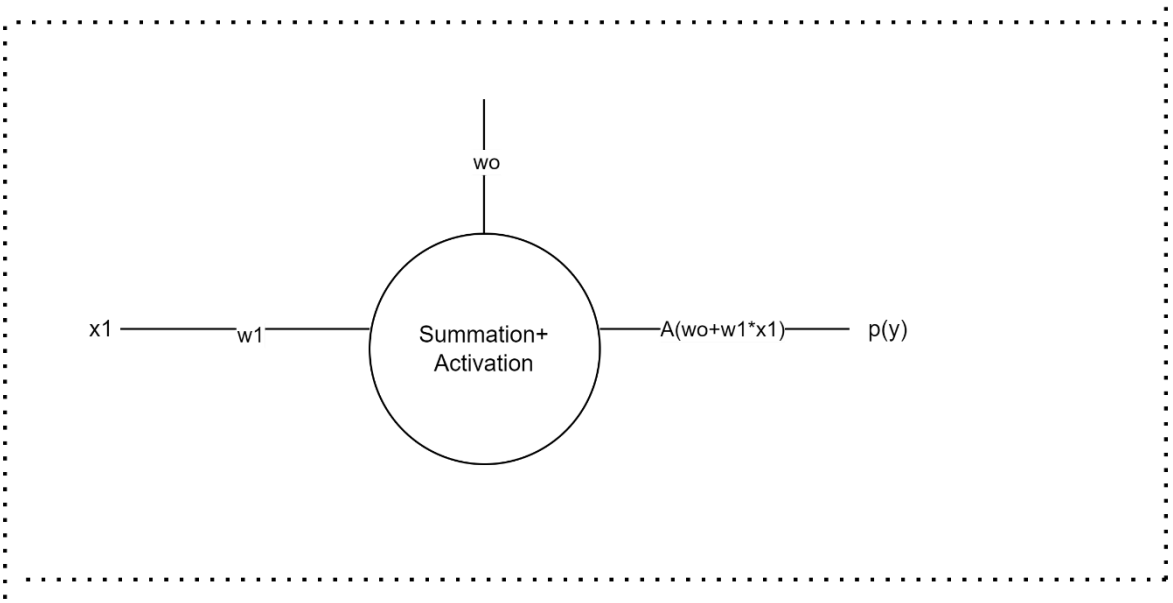
$$sigmoid\ is\ a\ function\ provides\ Non\ linearity$$

$sigmoid\ curve\ looks\ like\ Non\ linear$

$sigmoid, softmax,\ tanh, relu\ ===== > Non\ linearity$

$all\ these\ functions\ called\ Activation\ Functions$:

- $sigmoid\ ====== Binary\ classification$
- $softmax\ ====== Multi\ classification$
- tanh $tanh\ ======== Classsification$
- $ReLU: Rectified\ Linear\ Unit : Regression\ problem$



Neuron= Summation + Activation

In any Neural network there 3 layers available

1) *Input layer*
2) *Hidden layer*
3) *Output layer*

Input layer vs Output layer

$Age(x_1)$ $Gender\ (x_2)$ $Income\ (x_3)$ $Loan\_status(y)$

Input layer:

$Number\ of\ inputs = Number\ of\ neurons\ in\ the\ Input\ layers$

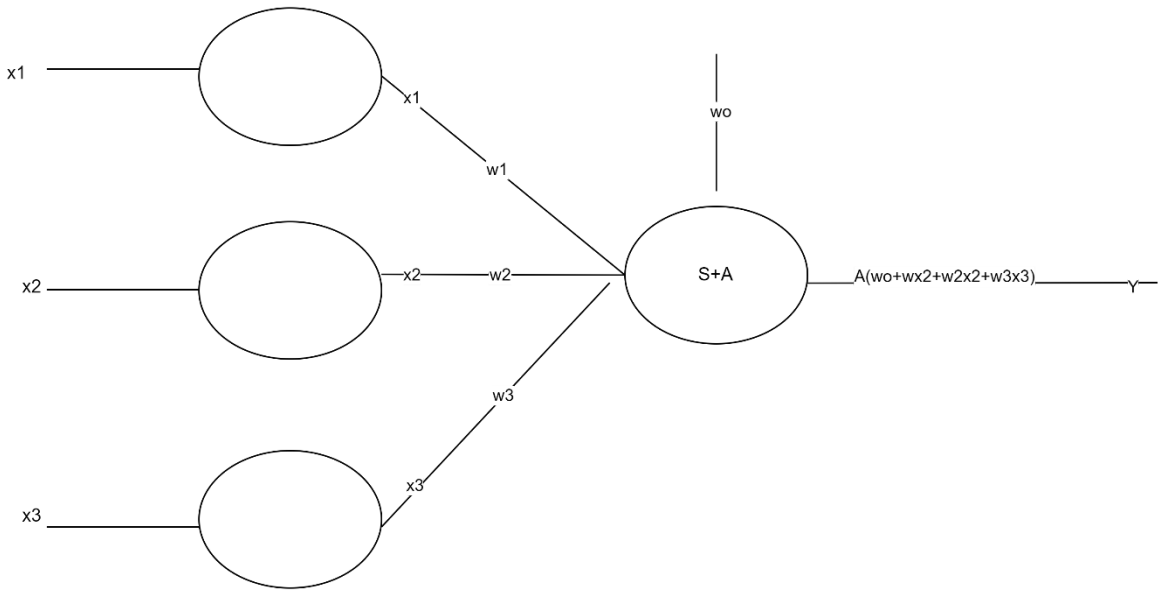
Ex: $Number\ of\ inputs = 3; Number\ of\ neurons = 3$

Output layers:

If it is binary classification: 1 *Neuron is enough*

$If\ it\ is\ Multi\ classification$:

$Number\ of\ Neurons\ =\ Number\ of\ lables$



Hidden layer : Model complexity

$Input\ layer$

$Hidden\ layer\ -\ 1:\ \ How\ to\ choose\ neurons\ ?$

$Hidden\ layer\ -\ 2$

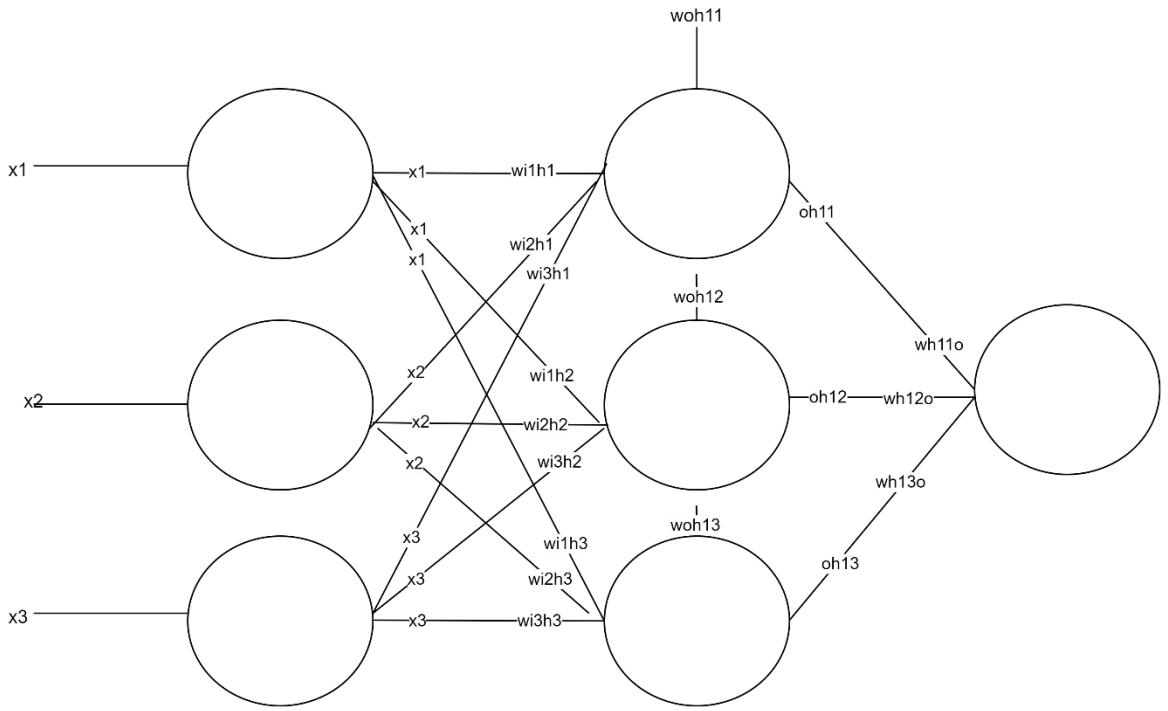
$Hidden\ layer\ -\ 3$

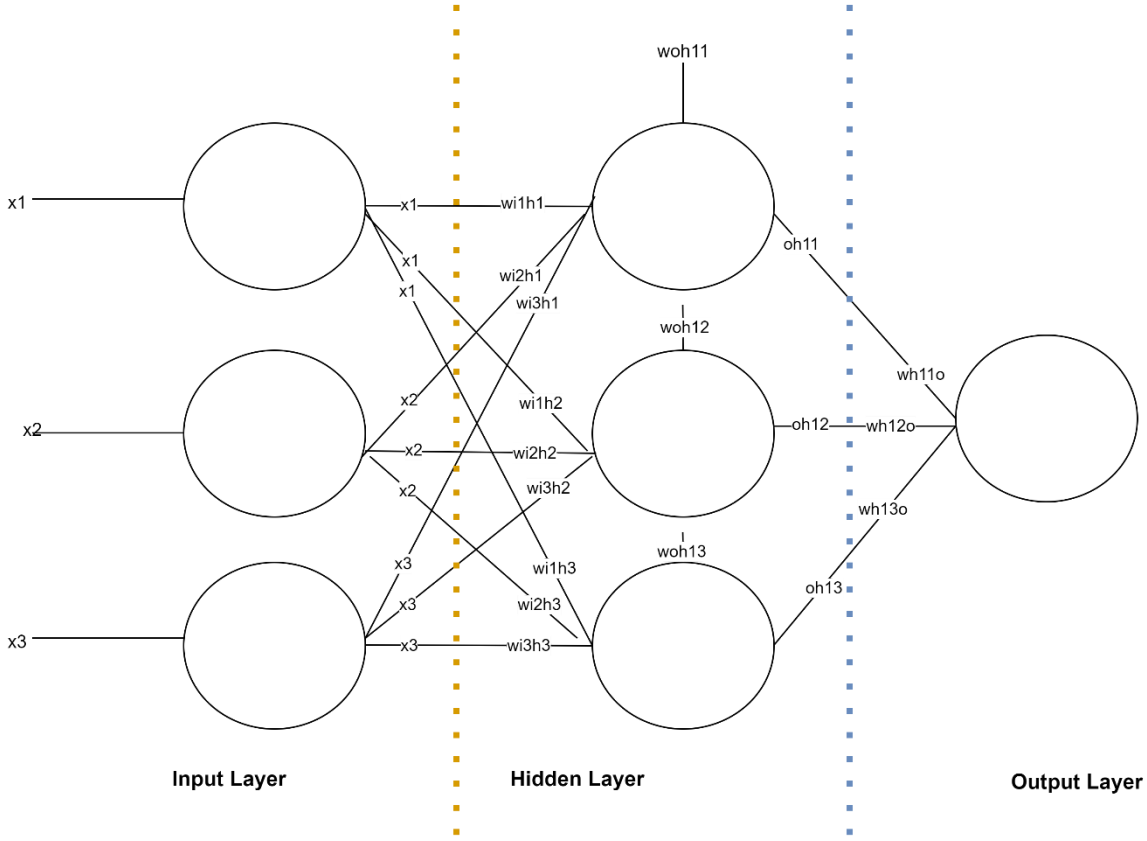
$Output\ layer$

$Input\ layer: 3N$

$Hidden\ layer\ -\ 1:\ 3\ Neurons$

$Output\ layer: 1N$





Input layer : *Just Pass the Inputs*

Hidden layer weights

$$w_{i1h1} = Input\ layer\ Neuron - 1\ to\ Hidden\ layer\ of\ Neuron - 1$$

$$w_{i1h2} = Input\ layer\ Neuron - 1\ to\ Hidden\ layer\ of\ Neuron - 2$$

$$w_{i1h3} = Input\ layer\ Neuron - 1\ to\ Hidden\ layer\ of\ Neuron - 3$$

$$w_{i2h1} = Input\ layer\ Neuron - 2\ to\ Hidden\ layer\ of\ Neuron - 1$$

$$w_{i2h2} = Input\ layer\ Neuron - 2\ to\ Hidden\ layer\ of\ Neuron - 2$$

$$w_{i2h3} = Input\ layer\ Neuron - 2\ to\ Hidden\ layer\ of\ Neuron - 3$$

$$w_{i3h1} = Input\ layer\ Neuron - 3\ to\ Hidden\ layer\ of\ Neuron - 1$$

$$w_{i3h2} = Input\ layer\ Neuron - 3\ to\ Hidden\ layer\ of\ Neuron - 2$$

$$w_{i3h3} = Input\ layer\ Neuron - 3\ to\ Hidden\ layer\ of\ Neuron - 3$$

$$w_{oh11} = Bias\ of\ Hidden\ layer - 1\ of\ Neuron - 1$$

$$w_{oh12} = Bias\ of\ Hidden\ layer - 1\ of\ Neuron - 2$$

$$w_{oh13} = Bias\ of\ Hidden\ layer - 1\ of\ Neuron - 3$$

$$Number\ of\ parameters\ to\ Train\ at\ Hidden\ layer - 1:$$

$$weights + Bias = 9 + 3 = 12params$$

$$Input\ layer\ neurons\ *\ Hidden\ layer\ Neurons + Total\ Number\ of\ Hidden\ layer\ Neurons$$

$$Hidden\ layer\ outputs:$$

$$o_{h11} = output\ of\ Hiddenlayer - 1\ of\ Nueron - 1$$

$$o_{h12} = output\ of\ Hiddenlayer - 1\ of\ Nueron - 2$$

$$o_{h13} = output\ of\ Hiddenlayer - 1\ of\ Nueron - 3$$

$$o_{h11} = Activation(w_{oh11} + w_{i1h1} * x_1 + w_{i2h1} * x_2 + w_{i3h1} * x_3)$$

$$o_{h12} = Activation(w_{oh12} + w_{i1h2} * x_1 + w_{i2h2} * x_2 + w_{i3h2} * x_3)$$

$$o_{h13} = Activation(w_{oh13} + w_{i1h3} * x_1 + w_{i2h3} * x_2 + w_{i3h3} * x_3)$$