**Deep Learning**

1. ANN: Artificial Neural network ( DNN)

All the kind of ML problems we can achieve using DNN

1. CNN: Convolution Neural network

All the kind of Image problems

Object detection

Video annotation

1. RNN: Recurrent Neural Network

Related To NLP problems

If you want to understand about Neural networks

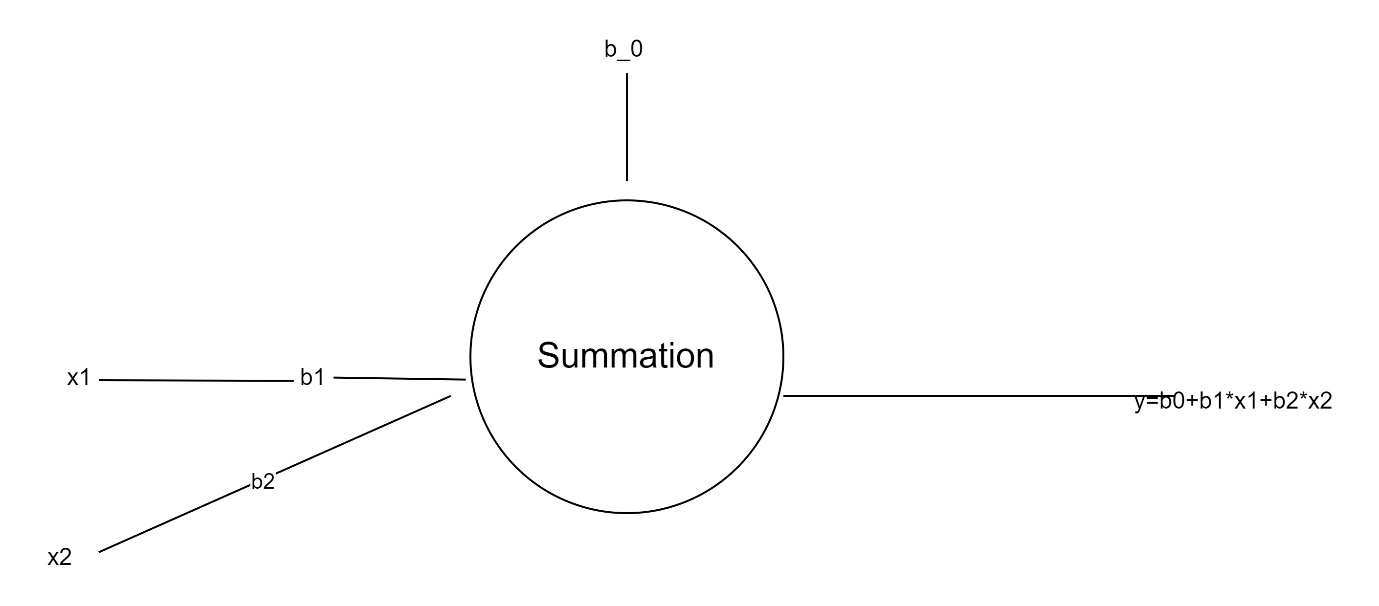
You need to understand about Neuron

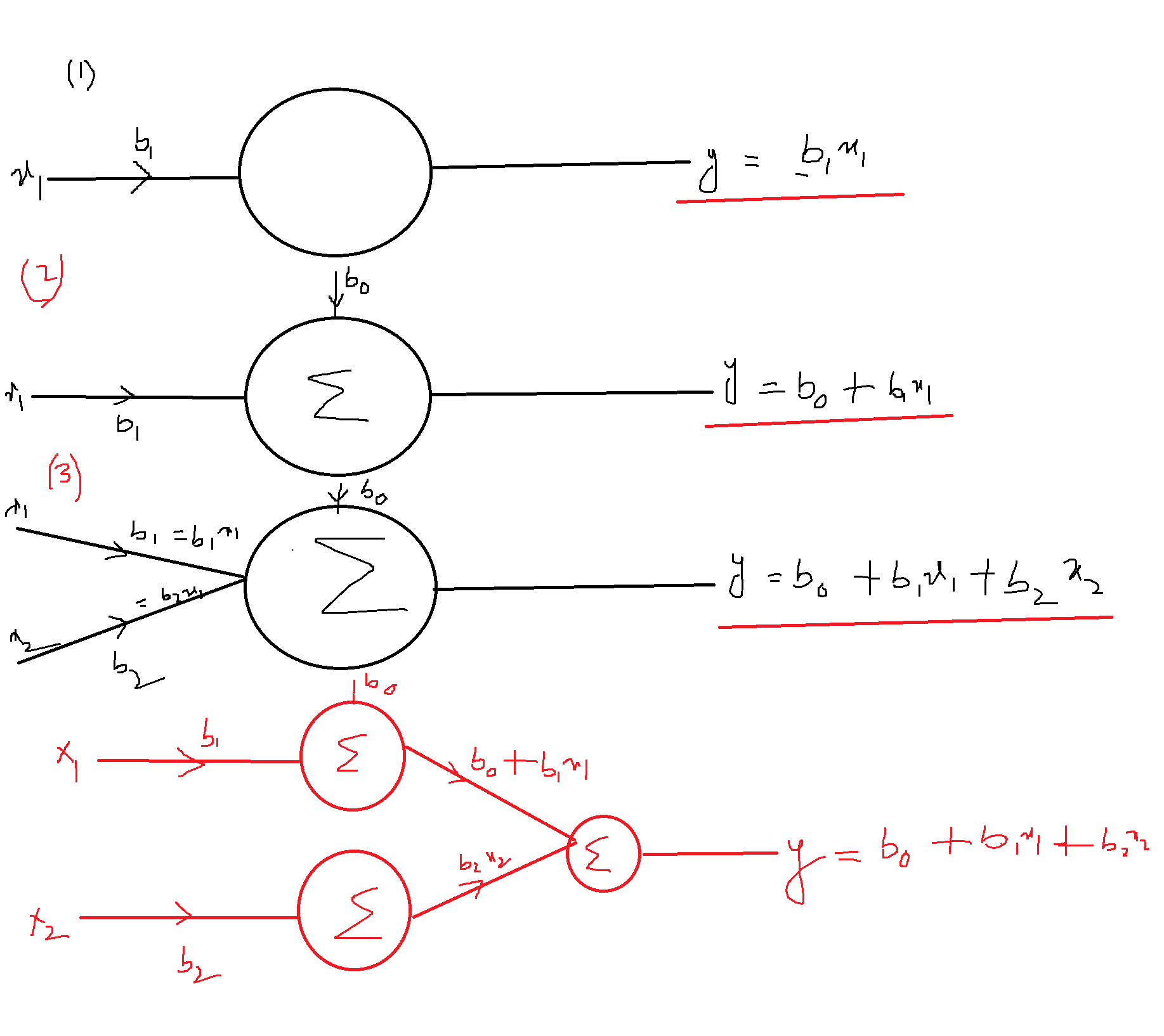
What’s happening inside the Neuron

Linear regression help

Y= bo+b1\*x1+b2\*x2+b3\*x3+…… +bn\*xn

Y= b1\*x1





If you observe in the above neurons, we are just passing the inputs

It is kind of Linear combination of inputs

B1\*x1+b2\*x2+b3\*x3 ========

Our model will not identify the patterns

Imagine our brain : So many neurons will available

Each neuron connect with other neurons in such a way, even a complex problem also

Our brain try to get the solution

We need to enhance the neuron, with Non linear property

There two types maths:

1. Linear maths

Passing the output

1. Non linear maths

It is able to understand the pattern

Imagine Logistic Regression Problem

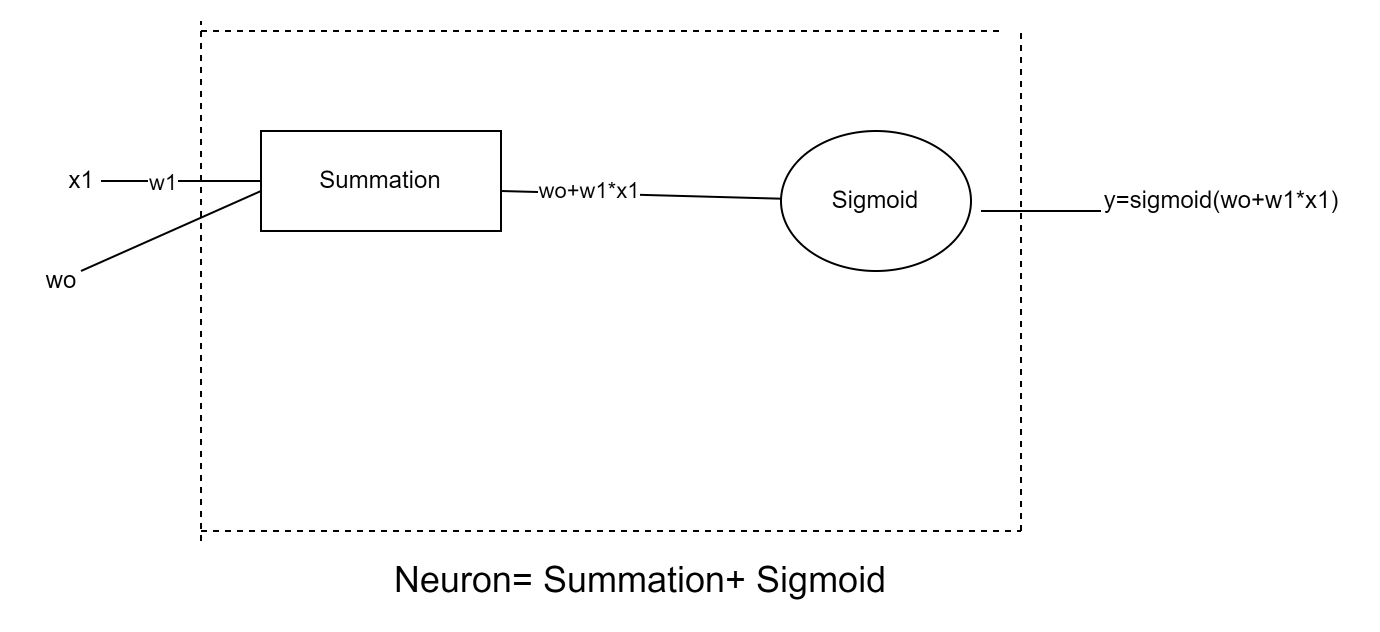
Step-1: We have input let assume we have input : x1

Step-2: We have one weight for x1: w1

Step-3: We have bias : w\_o

Step-4: Linear combination: wo+w1\*x1

Step-5: Sigmoid(LR)==== Sigmoid(wo+w1\*x1)



Sigmoid is a function: Non linear function

This is useful to Logistic regression problems

Binary classification problem

There are many Non linear function and Linear functions available

After computation will pass the output to the Non linear function, These Non linear function called

**Activation Function**

So sigmoid is one of the activation function

We have some activation function available for the specific use case probelems

* Sigmoid
* Softmax
* Tanh
* ReLU

Every activation function has different equation, provides a different use case

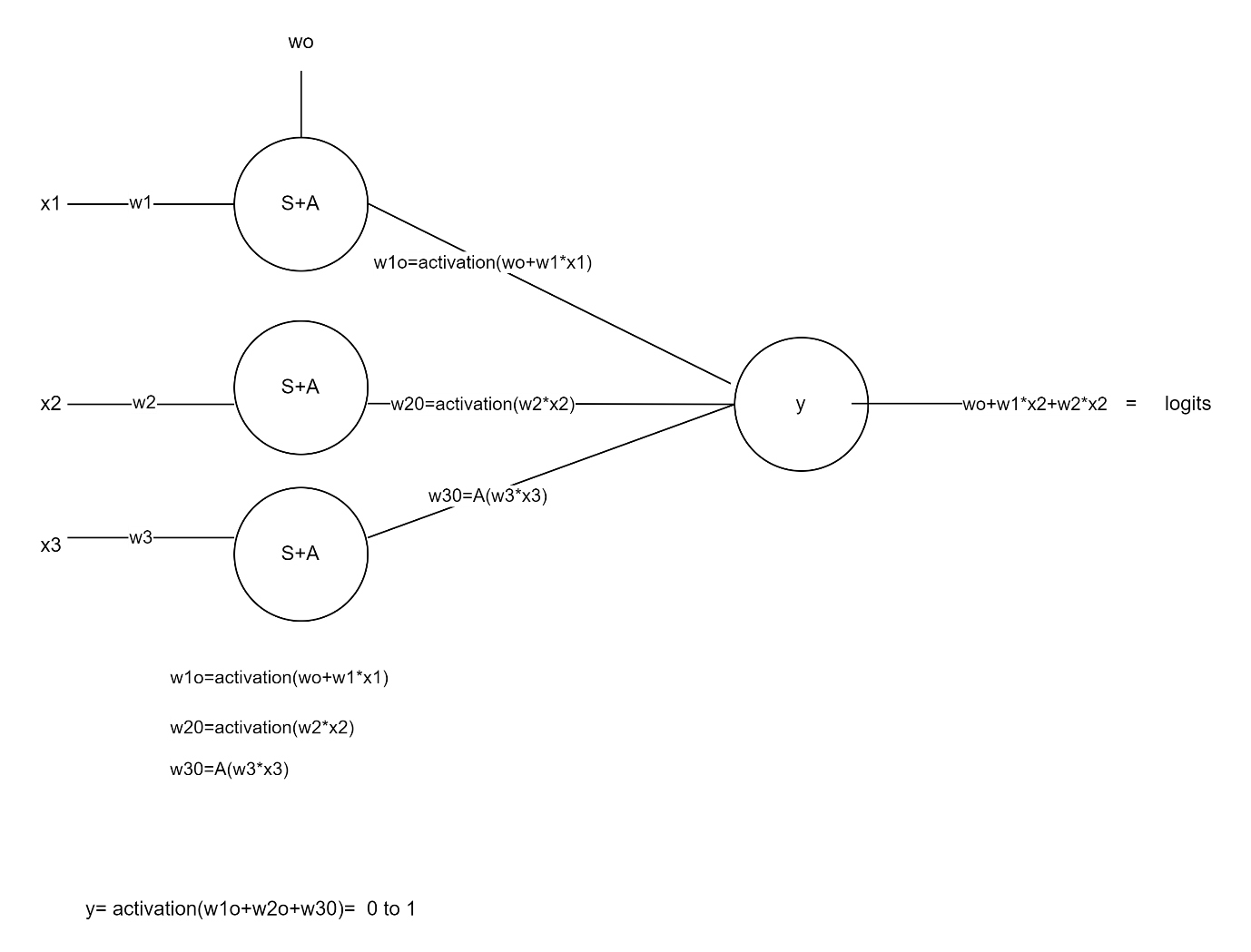
From the overall thing all Deep learning problems

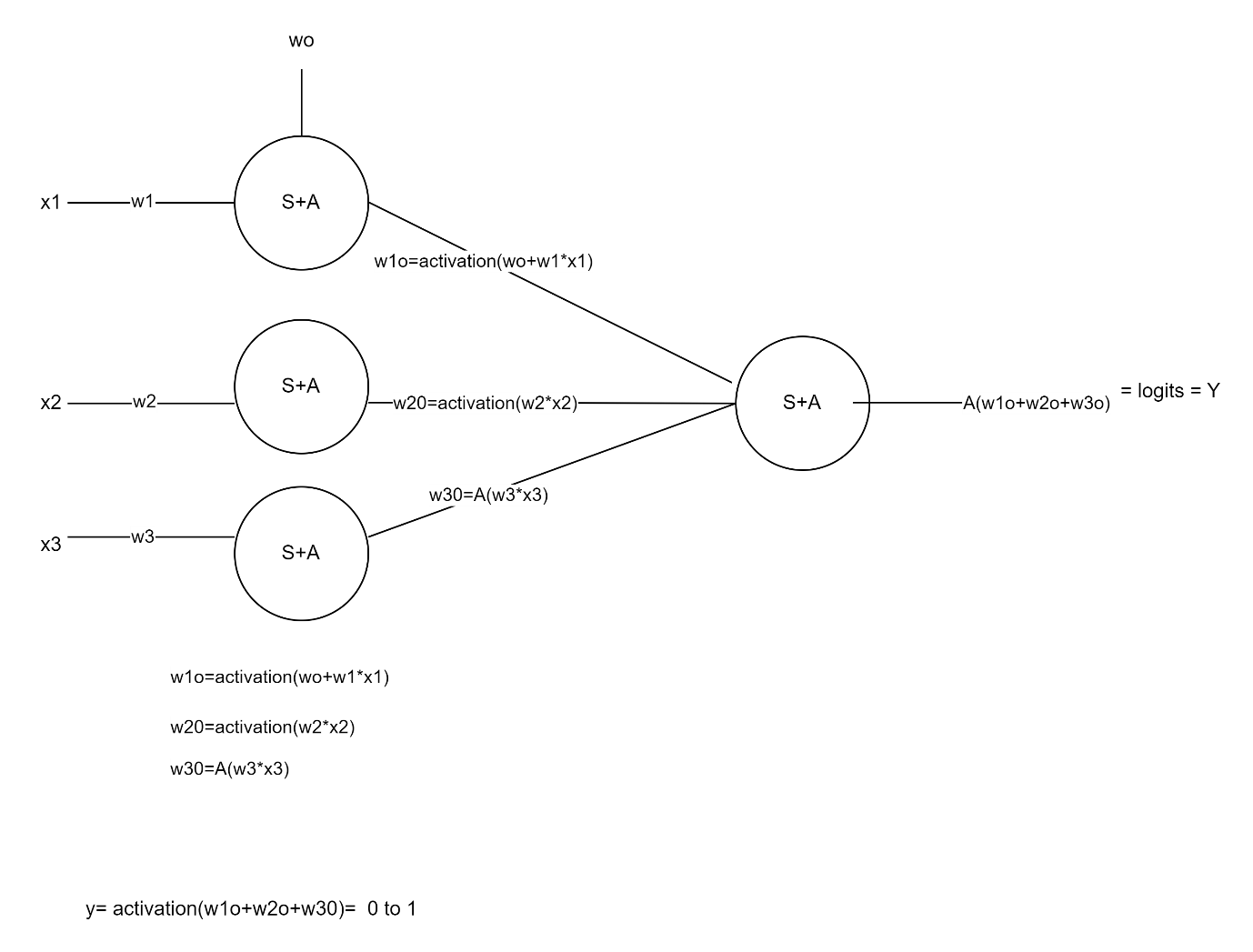
Is what happens inside the Neuron

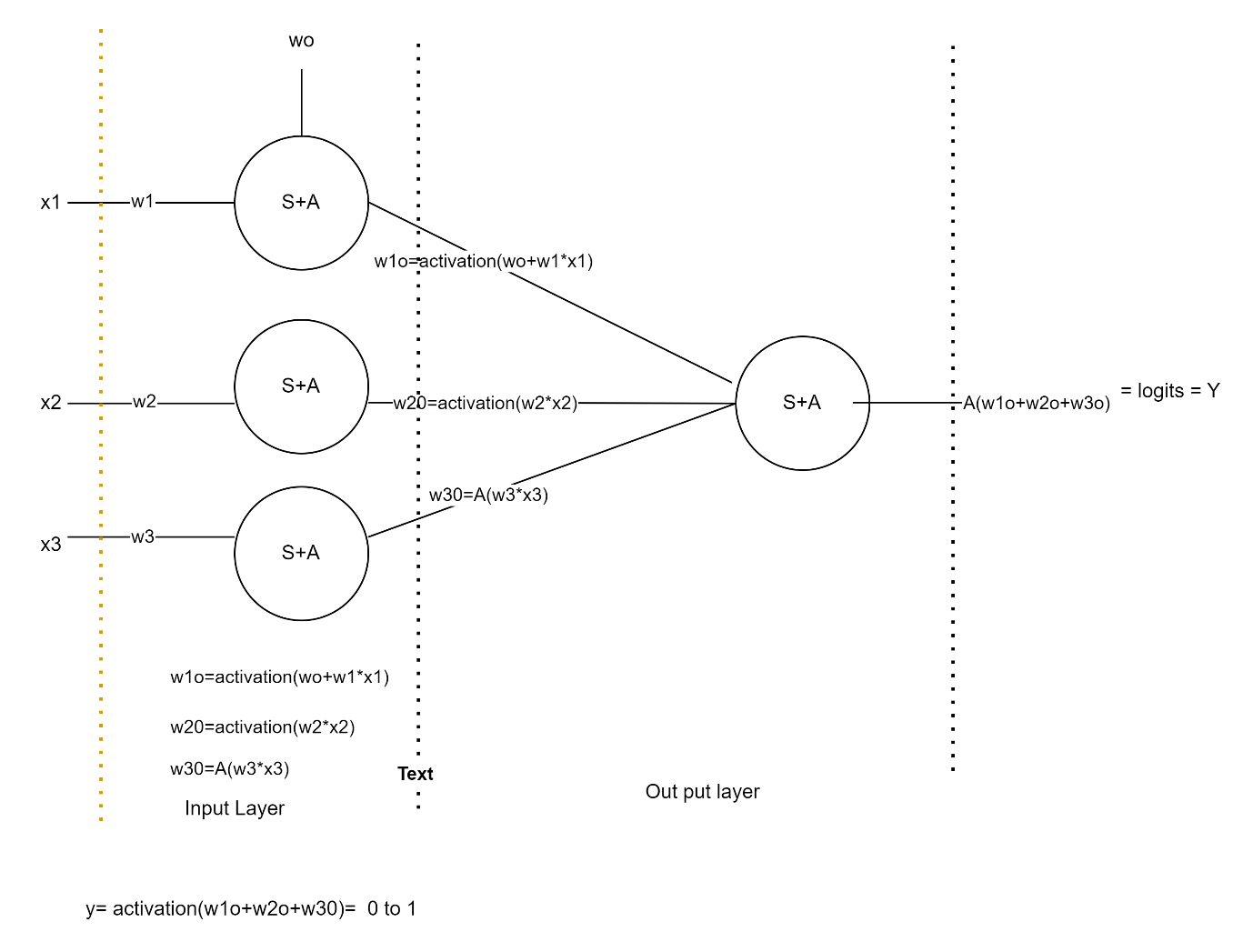
In side the neuron there two operation

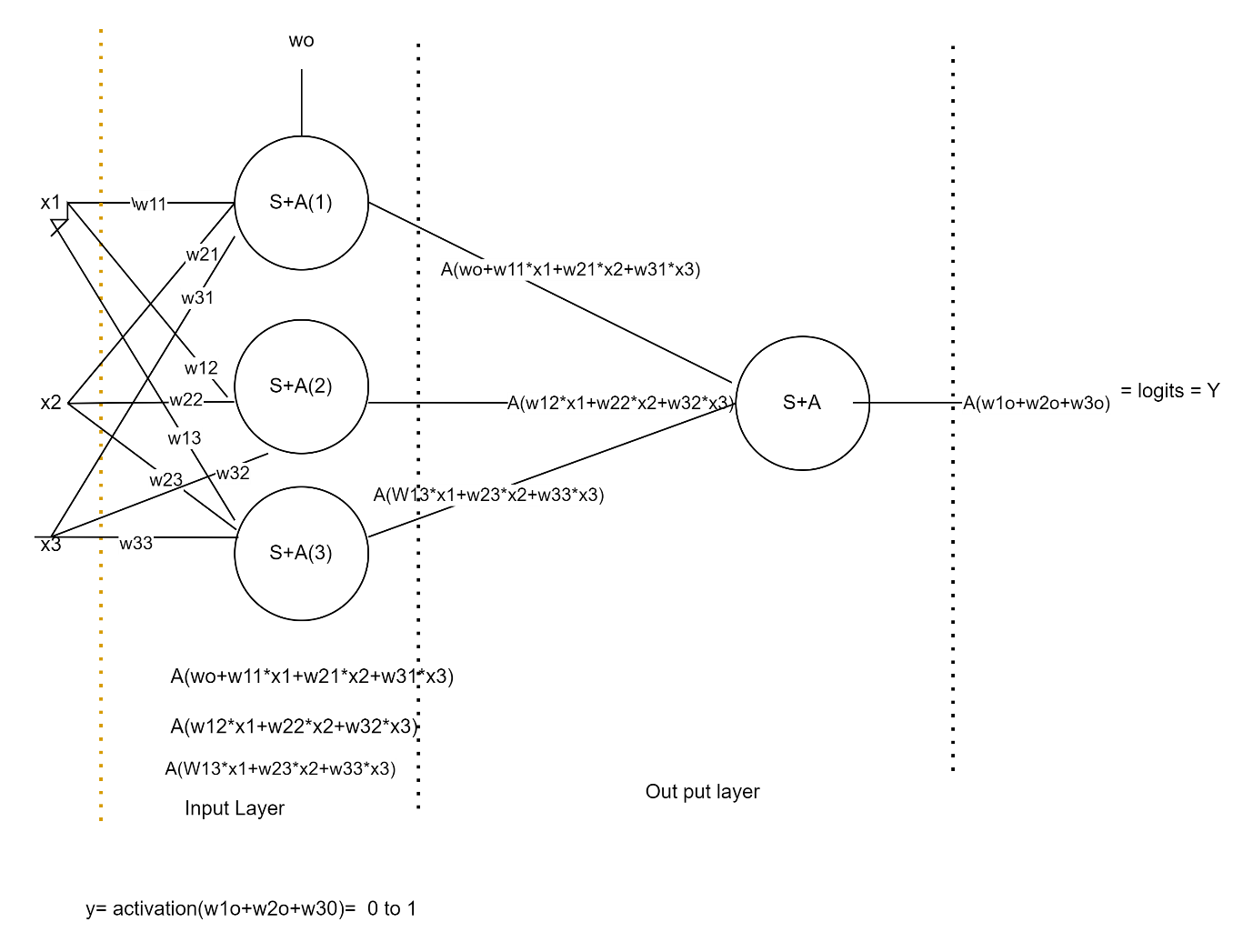
* Summation
* Activation

**Neuron-operation= Summation + Activation**







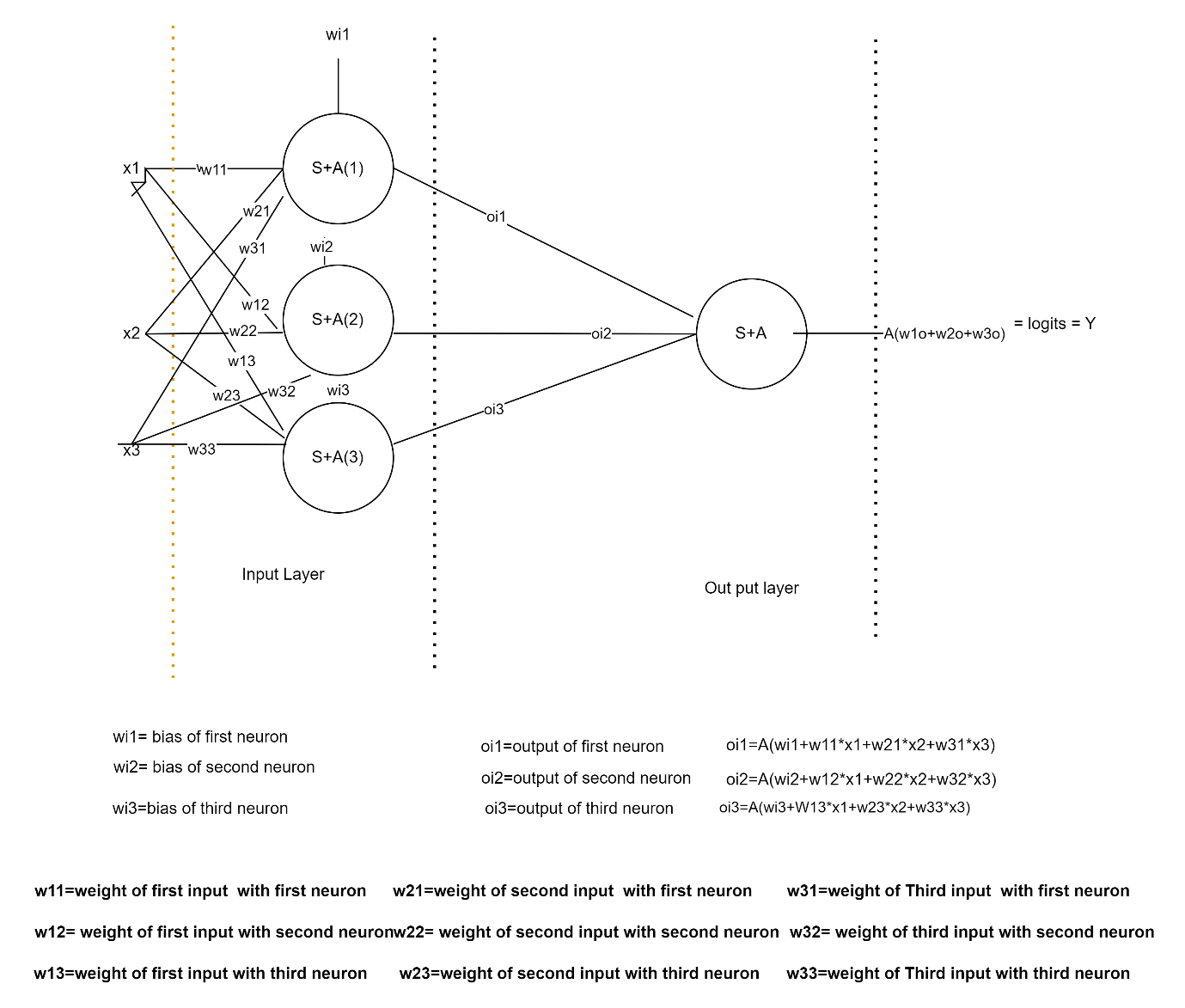


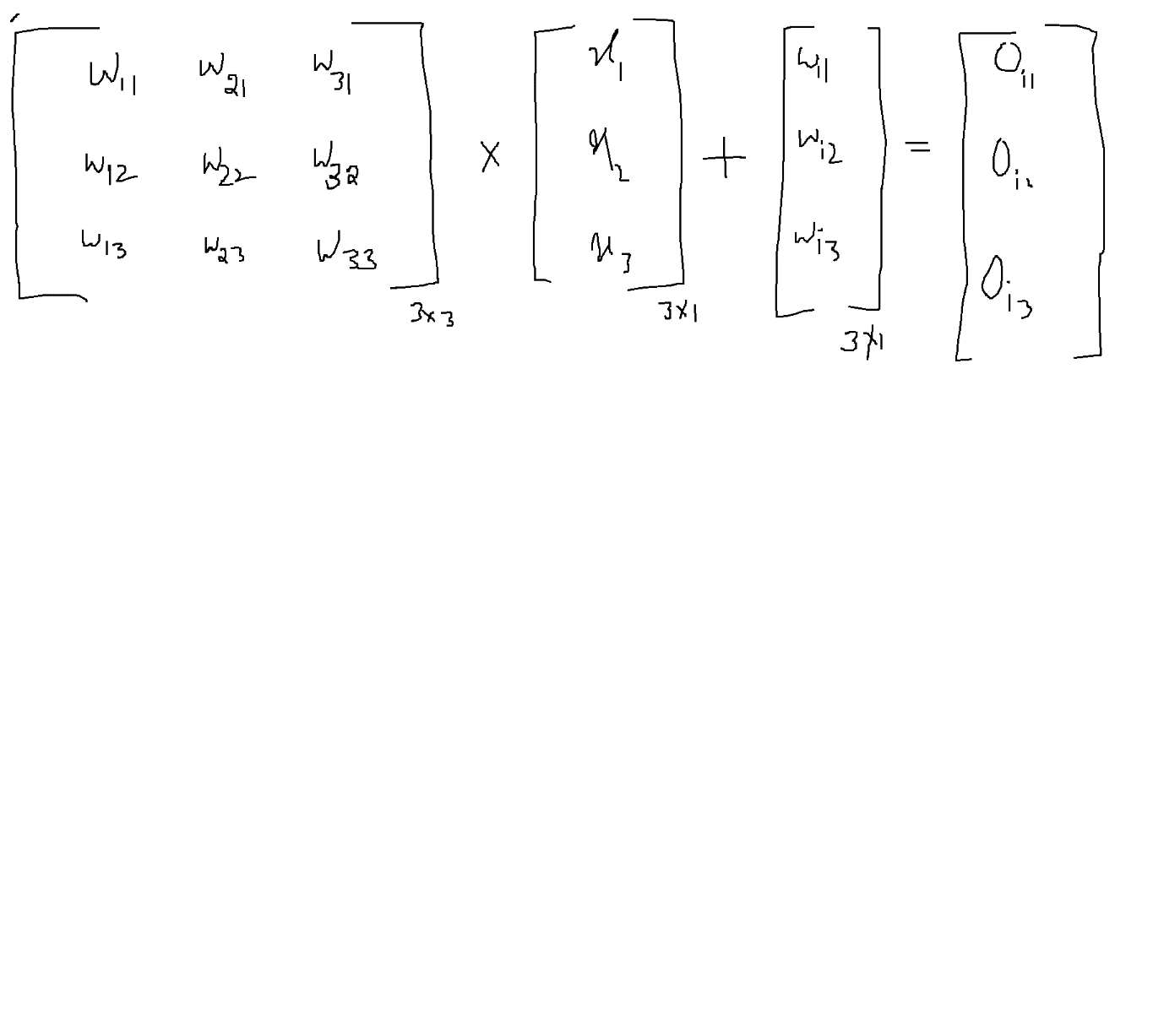
Final Conclusion

For the first layer

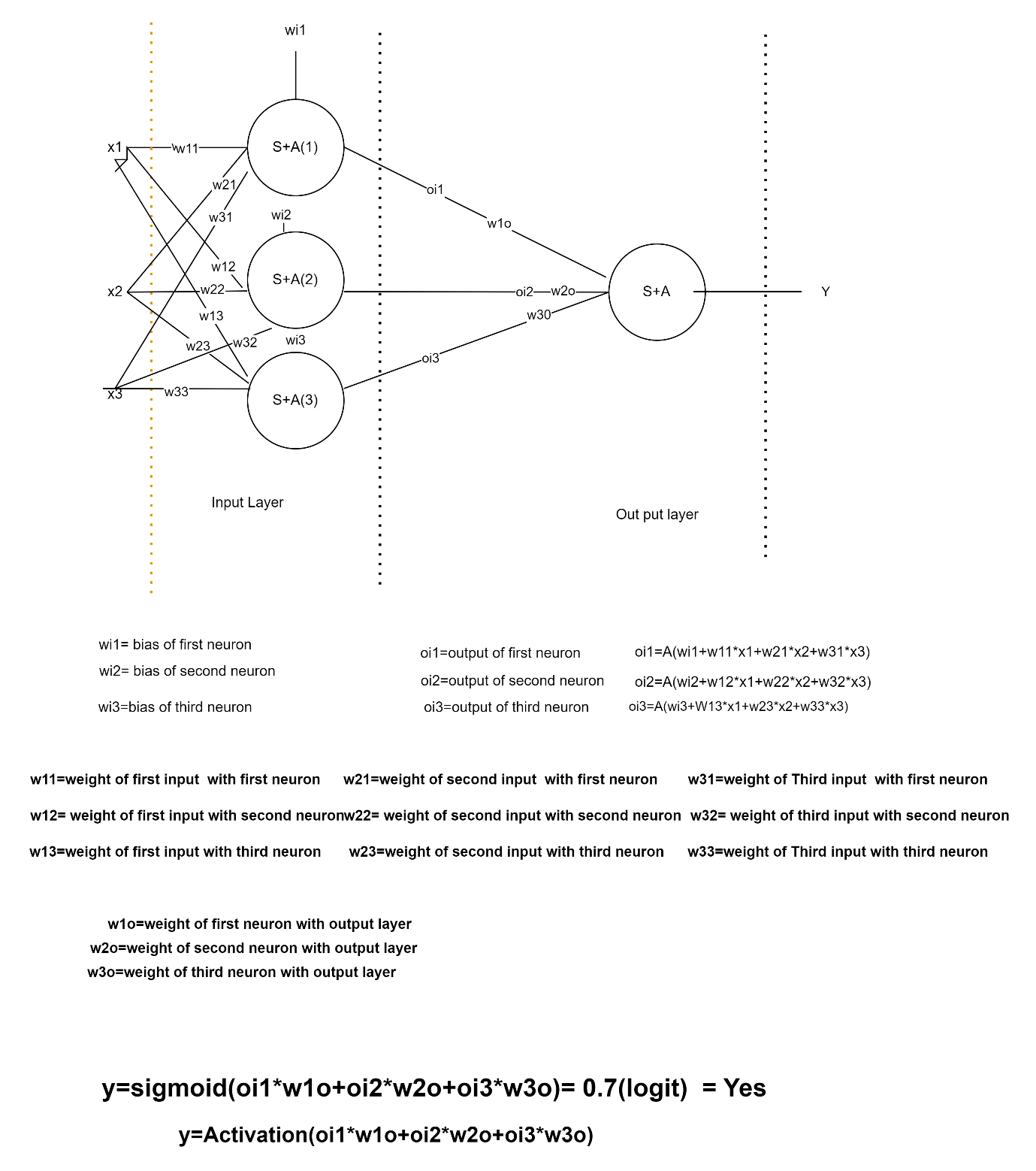
Number of coeffects , number of parameters to train is

3\*3+1= 9+1=10





Total weights to be train for input layer= 9+3=12



Output layer parameters are 3

Input layer parameters are = 9+3 =12

Total parameters or weights= 12+3=15

Input layer :

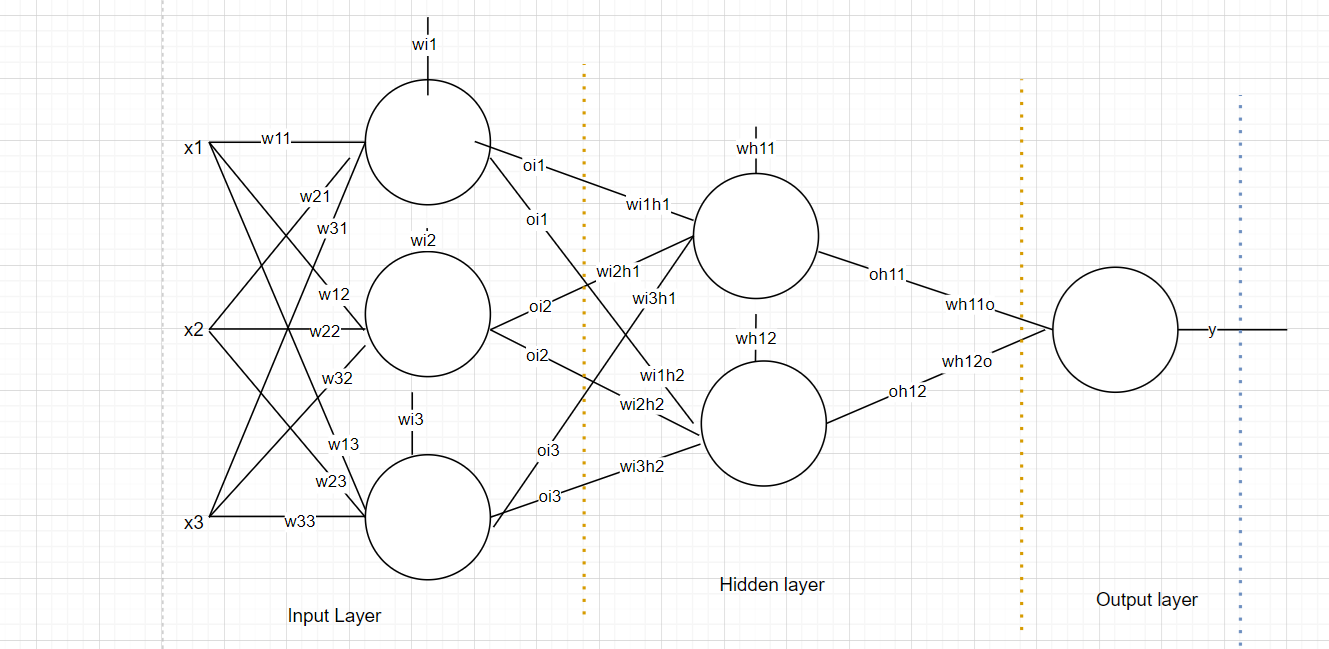
Neurons will depends on number of inputs

Hidden layer – 1: How many neurons: 2 neurons

Output layer:

Bi classification: 1 Neuron

**Hidden layer:**



**Weights= 6+2=8**

**Hidden layer has two neurons = 2 outputs**

**Output layer:**

**Output layer=2**

**Total weights : Input layer + Hidden layer+ Output layer**

**Input layer= 9+3=12**

**Hidden layer= 6+2 =8**

**Output layer= 2**

**12+8+2=22 weights need to be train**

