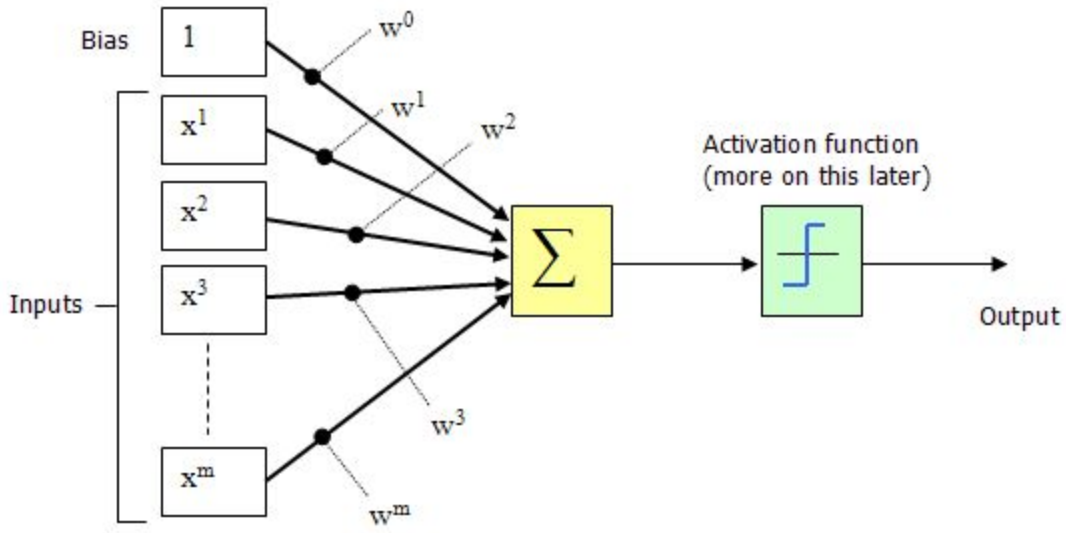
Why activation function? And its type?

The main motivation behind Artificial Neural Networks (ANNs for short) is to mimic how Biological Neural Networks (BNNs for short) works.



Here: f is activation function,

So,

$$output = f(bias + \Sigma weights * input)$$

Activation function **f** could be any function, linear or nonlinear. They bears special properties in neural networks.

1. Linear activation function like $f(x) = x$
2. Non linear activation function like $f(x) = 1/(1 - e^{-x})$

Reason 1

Lets say, three input,

$[a_1\ a_2\ a_3] = [2\ 1\ 3]$ and $[w_1\ w_2\ w_3] = [0.4\ 0.3\ 0.2]$ and bias=1

Thus, $b + \Sigma a_i * w_i$ =2.7 for i 1 to 3

Lets pull output for single neural having fed with linear and non linear activation function,

In linear activation, $f(x) = x$ leads to 2.7

and ,
Nonlinear activation,

$f(x) = 1/(1 - e^{-x})$ leads $f(2.7) = 0.937$

Thus: Non linear activation is used to map all input to certain bounded output. In above case, f(x)=x bound to $(-\infty, \infty)$ where as sigmoid function is bound to (0,1). When the range of the activation function is finite, gradient-based(derivative) training methods tend to be more stable

Reason 2

Activation function should be differentiable.
In above example,
Linear function $f'(x) = 1$
Non linear, $f'(x) = f(x)(1 - f(x))$
The differentiability property is associated with gradient learning optimization.

Then What is gradient Learning? And learning optimization?
A gradient measures how much the output of a function changes if you change the inputs a little bit.
Higher the gradient, the steeper the slope and the faster a model can learn. But if the slope is zero, the model stops learning. Said it more mathematically, a gradient is a partial derivative with respect to its inputs