Neural Networks are applied to many important real world problems like,

- image recognition
- translation

A SIMPLE NEURAL NET

- $a^{(0)}$ and $a^{(1)}$ are scalars
- $a^{(0)}$ is input
- $a^{(1)}$ is output

This relationship can be expressed:

$$a^{(1)} = \sigma(wa^{(0)} + b)$$

$a \Rightarrow$ "activity"

$w \Rightarrow$ "weight"

$b \Rightarrow$ "bias"

$\sigma \Rightarrow$ "activation function"

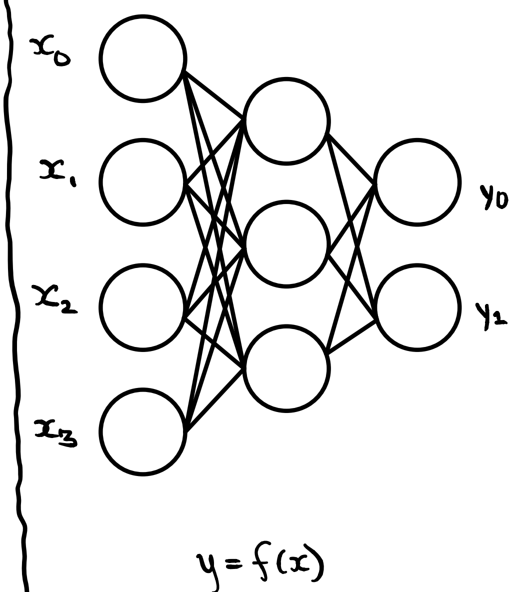


DIAGRAM OF A NEUR-L NETWORK

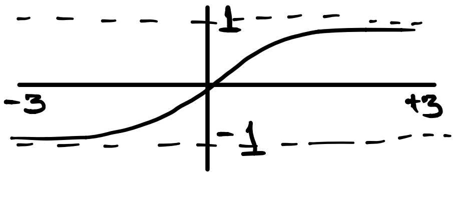
- circles are neurons
- lines are the network of connections between them.

The reason why sigma σ departs from roman lettering is to signify it's etymology - sigma is what makes neural networks similar to the brain.

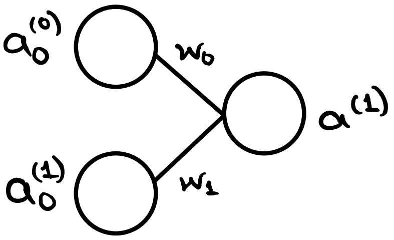
Neurons recieve information from their neighbours, through both chemical and electrical stimulation. If sum of these stimulations exceed a certain threshold - the neuron is activated and will begin stimulating its neighbours.

One function which acts similar to the neural stimulation function is the hyperbolic tanh(x).

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



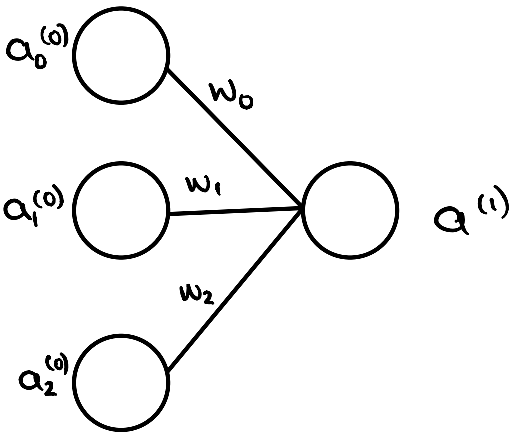
tanh(x) is of the sigmoid family of functions - which is why σ is used.



The output of this neural net is the activation of $a^{(1)}$. That activation can be expressed:

$$a^{(1)} = \sigma(w_0 a_0^{(0)} + w_1 a_1^{(0)} + b)$$

This three input neural net can be expressed:



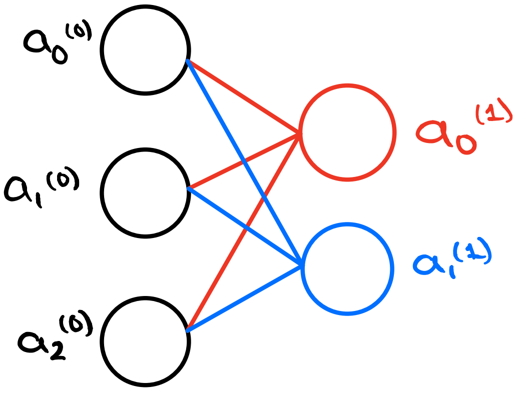
$$a^{(1)} = \sigma(w_0 a_0^{(0)} + w_1 a_1^{(0)} + w_2 a_2^{(0)} + b)$$

The combination of these inputs can be generalized.

$$a^{(1)} = \sigma\left[\left(\sum_{j=0}^{n} w_j a_j^{(0)}\right) + b\right],$$ as a summation

$$= \sigma(w \cdot a^{(0)} + b)$$

This same generalization can be applied to the outputs.



Rudimentarily this could be written

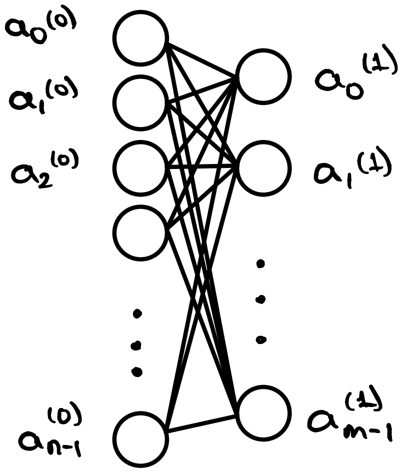
$$a_0^{(1)} = \sigma(w_0 \cdot a^{(0)} + b_0)$$

$$a_1^{(1)} = \sigma(w_1 \cdot a^{(0)} + b_1)$$

However these can be combined again.

$$a^{(1)} = \sigma(\underbrace{W^{(1)}}_{\text{weight matrix}} \cdot a^{(0)} + \underbrace{b^{(1)}}_{\text{combined bias vector.}})$$

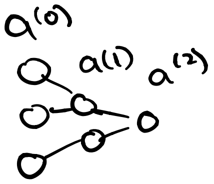
A generalization of a two layer network,



$$a^{(1)} = \sigma(W^{(1)} \cdot a^{(0)} + b^{(1)})$$

$$\begin{bmatrix} a_0^{(1)} \\ a_1^{(1)} \\ \vdots \\ a_{m-1}^{(1)} \end{bmatrix} = \sigma\left(\begin{bmatrix} w_{0,0}^{(1)} & w_{0,1}^{(1)} & \dots & w_{0,n-1}^{(1)} \\ w_{1,0}^{(1)} & w_{1,1}^{(1)} & \dots & w_{1,n-1}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m-1,0}^{(1)} & w_{m-1,1}^{(1)} & \dots & w_{m-1,n-1}^{(1)} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_{n-1}^{(0)} \end{bmatrix} + \begin{bmatrix} b_0^{(1)} \\ b_1^{(1)} \\ \vdots \\ b_{m-1}^{(1)} \end{bmatrix} \right)$$

Sometimes neural nets have hidden layers,



$$a^{(1)} = \sigma(W^{(1)} \cdot a^{(0)} + b^{(1)})$$

$$a^{(2)} = \sigma(w^{(2)} \cdot a^{(1)} + b^{(2)})$$

Generalized as,

$$a^{(L)} = \sigma(W^{(L)} \cdot a^{(L-1)} + b^{(L)})$$

This algebra is required for feed forward neural net.