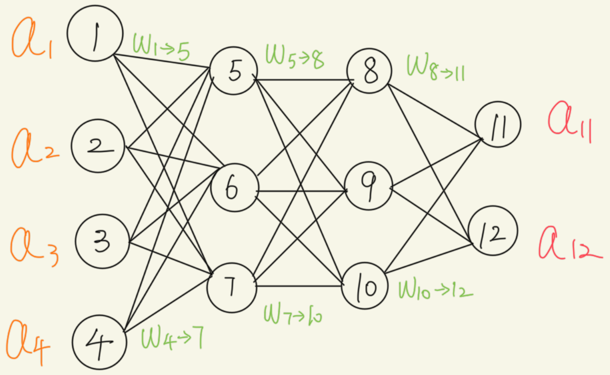
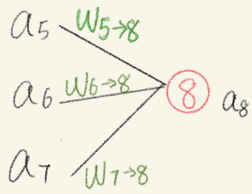
Feed Forward Neural Networks

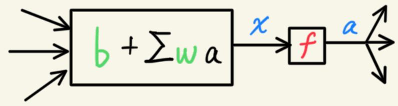


input dimension: 4

output dimension: 2

hidden layers: [3, 3]





$$x_8 = b_8 + \sum_{i=5}^{7} a_i W_{i\to8}$$

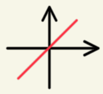
$$a_8 = f_8(x_8)$$

Computation feeds forward:

- set inputs
- loop through layers in the network
  - compute each node's activation (and store it!)
    - uses prev. layer's activations

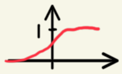
Activation Functions

linear: $a = x$



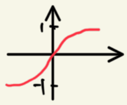
$\frac{da}{dx} = 1$

sigmoid: $a = \frac{1}{1+e^{-x}}$



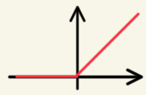
$\frac{da}{dx} = a(1-a)$

tanh: $a = \frac{1+e^{-2x}}{1-e^{-2x}}$



$\frac{da}{dx} = 1-a^2$

ReLU: $a = Max(0, x)$



$\frac{da}{dx} = \begin{cases} 0 & a=0 \\ 1 & a>0 \end{cases}$

Why we use the hidden layers?

- linear activations: theorem 1

  Does Nothing (nothing difference with one single neuron)

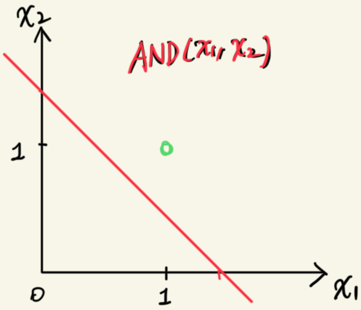
  If we compute some linear functions with a neuron, then pass it through some other linear functions, we could have achieved same thing with one single neuron (with diff. weights).

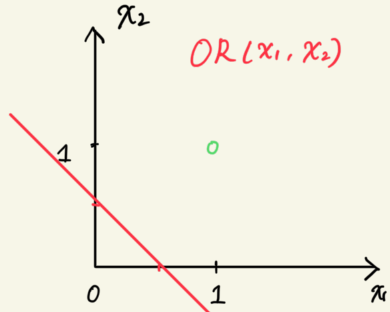
- non-linear activations: theorem 2

  Computes Anthing

Theorm 1: Let $f(\vec{x})$ and $g(\vec{x})$ be linear functions, then $f(g(\vec{x}))$ is linear too.

Theorm 2: If we can build the functions AND / OR / NOT, then we can compose them to represent any Boolean function.





AND(0,0) = 0
AND(1,0) = 0
AND(0,1) = 0
AND(1,1) = 1

OR(0,0) = 0
OR(1,0) = 1
OR(0,1) = 1
OR(1,1) = 1

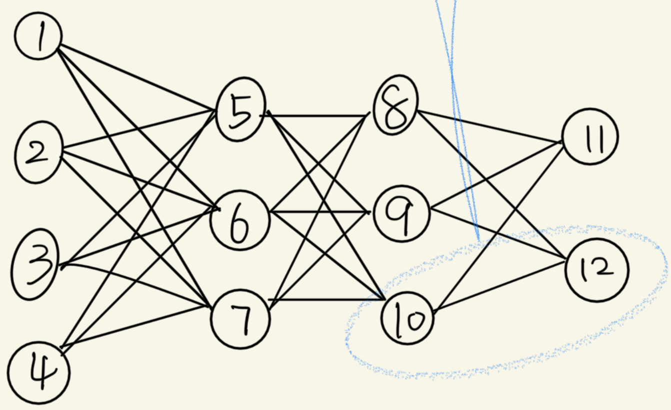
Gradient Descent Training

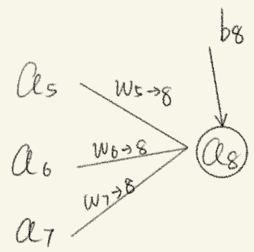
For each data point, the loss function needs to aggregate over output dimensions.

$$\mathcal{L}^j(\theta,\ data^j) = \sum_{o \in output}(t_o^j - a_o^j)^2 \qquad \text{Sum over outputs}$$

$$\mathcal{L}(\theta,\ dataset) = \frac{1}{N}\sum_{j=1}^{N} L^j(\theta,\ data^j) \qquad \text{Mean over data}$$

$$\theta = \begin{bmatrix} w_{1\to 5} \\ w_{1\to 6} \\ w_{1\to 7} \\ \vdots \\ w_{10\to 12} \\ b_5 \\ b_6 \\ \vdots \\ b_{11} \\ b_{12} \end{bmatrix} \qquad \nabla \mathcal{L} = \begin{bmatrix} \partial L/\partial w_{1\to 5} \\ \partial L/\partial w_{1\to 6} \\ \partial L/\partial w_{1\to 7} \\ \vdots \\ \partial L/w_{10\to 12} \\ \partial L/\partial b_5 \\ \vdots \\ \partial L/\partial b_{11} \\ \partial L/\partial b_{12} \end{bmatrix}$$





$$x_8 = b_8 + \sum_{i=5}^{7} a_i w_{i\to 8}$$