Non-Linear Classification:

If there are two many feature,
Then for. $x_1 \ldots x_n$ features.

if we want to add all the quadratic functions, then we need to pass $\frac{n(n+1)}{2} \approx \frac{n^2}{2}$ arguments to sigmoid function. This is computateonally so difficult.

Example: In computer vision.

We have $50 \times 50 = 2500$ pixels greyscale pic

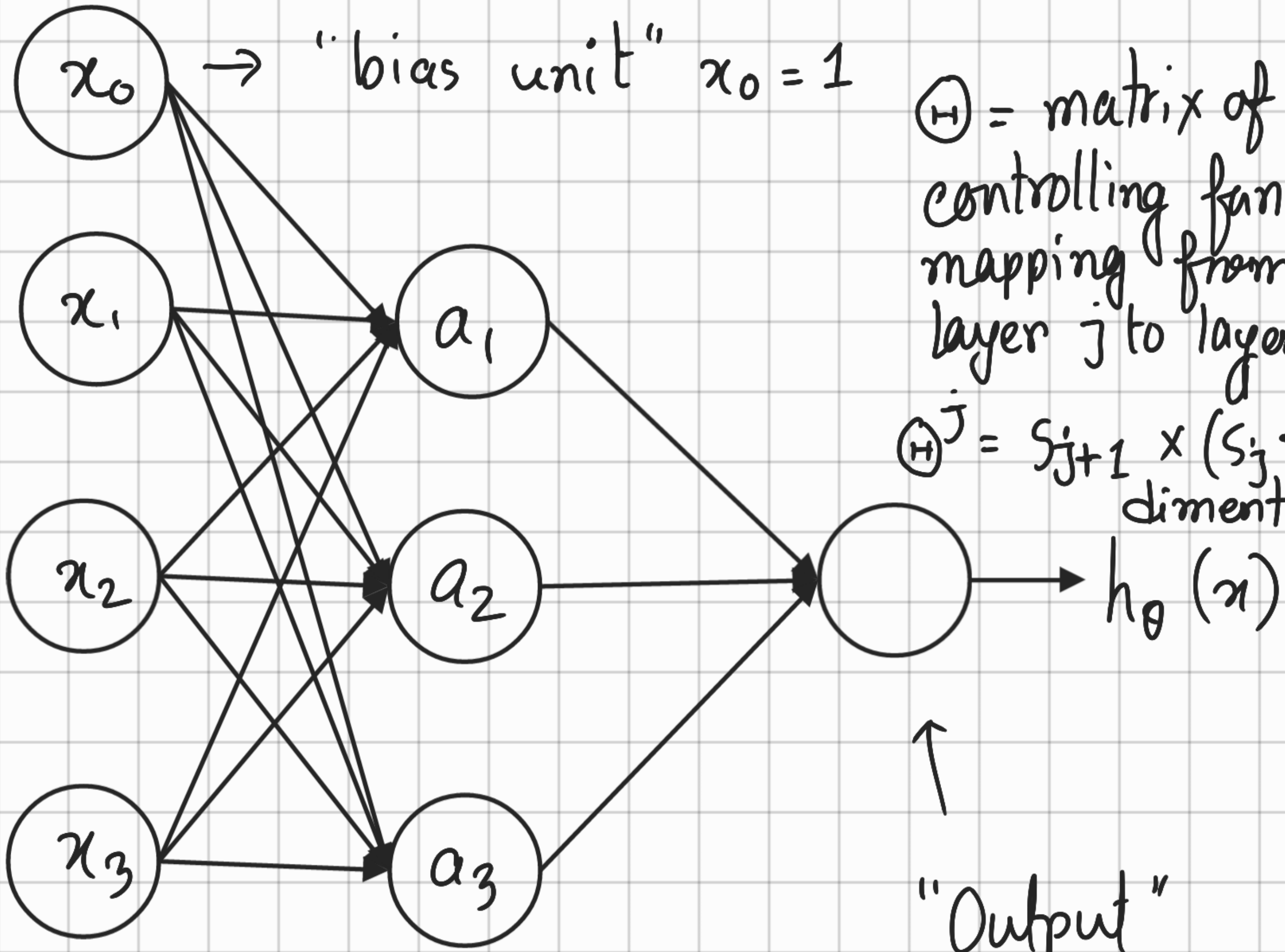
$$P = \begin{bmatrix} \text{pixel 1 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

$\approx$ 3 million features

So we need neural network.

Neural Network: tries to mimic brain.

Model Presentation:



→ "bias unit" $x_0 = 1$

Θ = matrix of weight controlling function mapping from layer j to layer j+1

$\Theta^{j} = s_{j+1} \times (s_j + 1)$ dimentional

↑ "Output"

↓ 3 units "input"

↓ hidden layer

a_i^{j} = "activation" of unit "i" in layer j

$$a_1^{(2)} = g\left(\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3\right)$$

$$g(z) = \frac{1}{1+e^{-z}}$$

⋮

$$h_\theta(x) = a_1^{(3)} = g\left(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)}\right)$$

Vectorized :

$$a^{(1)} = X$$

$$z^{(2)} = \Theta^{(1)} x = \Theta^{(1)} a^{(1)}$$

$$\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3 = z_1^{(2)}$$

$$a^{(2)} = g\left(z^{(2)}\right)$$

Add $a_0^{(2)} = 1$

$$z^{(i)} = a^{(i-1)}\, \Theta^{(i-1)T}$$

$$z^{(3)} = \Theta^{(2)} a^{(2)}$$

$$h_\Theta(x) = a^{(3)} = g\left(z^{3}\right)$$

x_1	x_2	$h_\theta(x)$
0	0	$g(-10) \approx 0$
0	1	$g(10) \approx 1$
1	0	$g(10) \approx 1$
1	1	$g(30) \approx 1$

$$h_\theta(x) = g\left(-10 + 20 x_1 + 20 x_2\right)$$

$$h_\theta(x) \approx x_1 \text{ OR } x_2$$