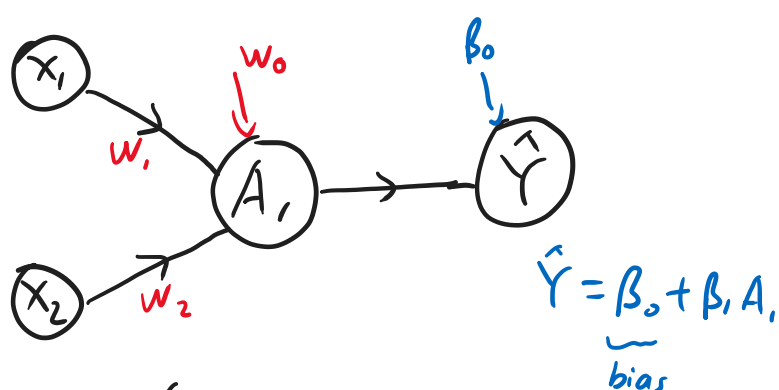


Simple Neural Network

$p=2$, 1 hidden layer, $K=2$ hidden nodes



$$A_1 = g(w_0 + w_1 x_1 + w_2 x_2)$$

bias

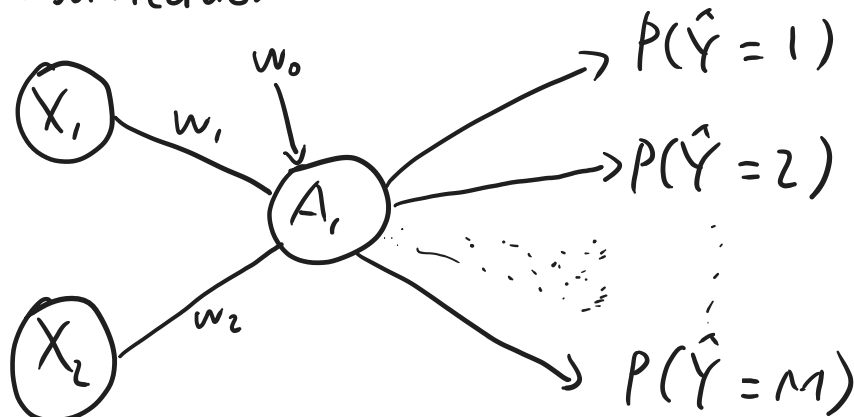
g = activation function

$$g(z) = z$$

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

$$g(z) = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases} \text{ (Relu)}$$

Classification



black box models

input: X

output: \hat{Y}

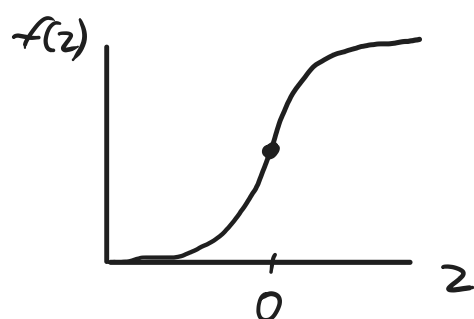
going from $X \rightarrow \hat{Y}$ complicated,
 have to estimate many parameters

Sigmoid $f(z) = \frac{e^z}{1+e^z}$

$$f(0) = \frac{1}{2}$$

$$z \rightarrow \infty, f(z) \rightarrow 1$$

$$z \rightarrow -\infty, f(z) \rightarrow 0$$



Simple Model (1 hidden layer)

- K (# of hidden nodes)

- g (activation function)

Input \rightarrow hidden layer(s) $\rightarrow \hat{Y}$

Complicated!

"Deep" Learning (more than 1 hidden layer)

CNN

input $\xrightarrow{\text{feature engineering}}$ convolutional layer $\xrightarrow{\text{keeping the key parts}}$ max pooling layer

\rightarrow convolutional layer \rightarrow max pooling layer

\rightarrow regular hidden layer $\rightarrow \hat{Y}$