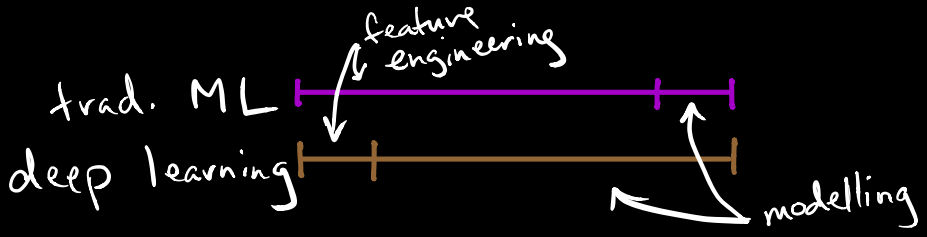
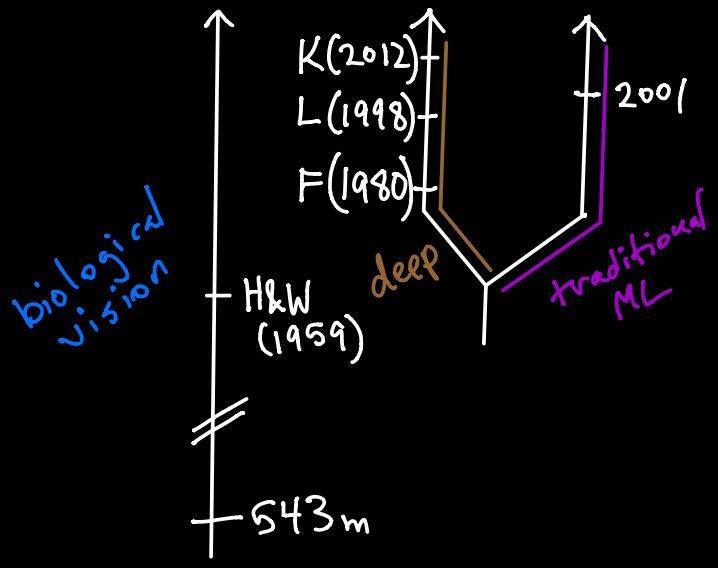
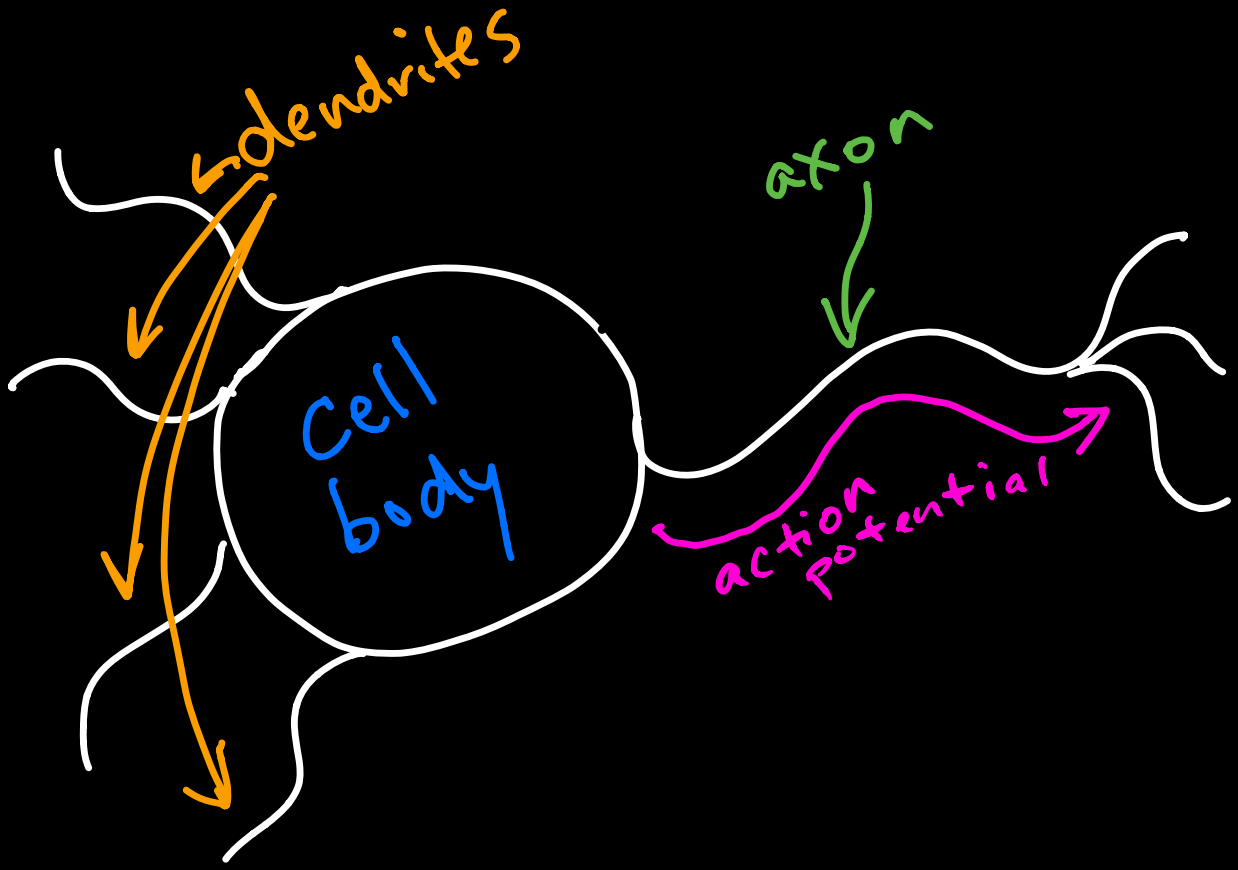
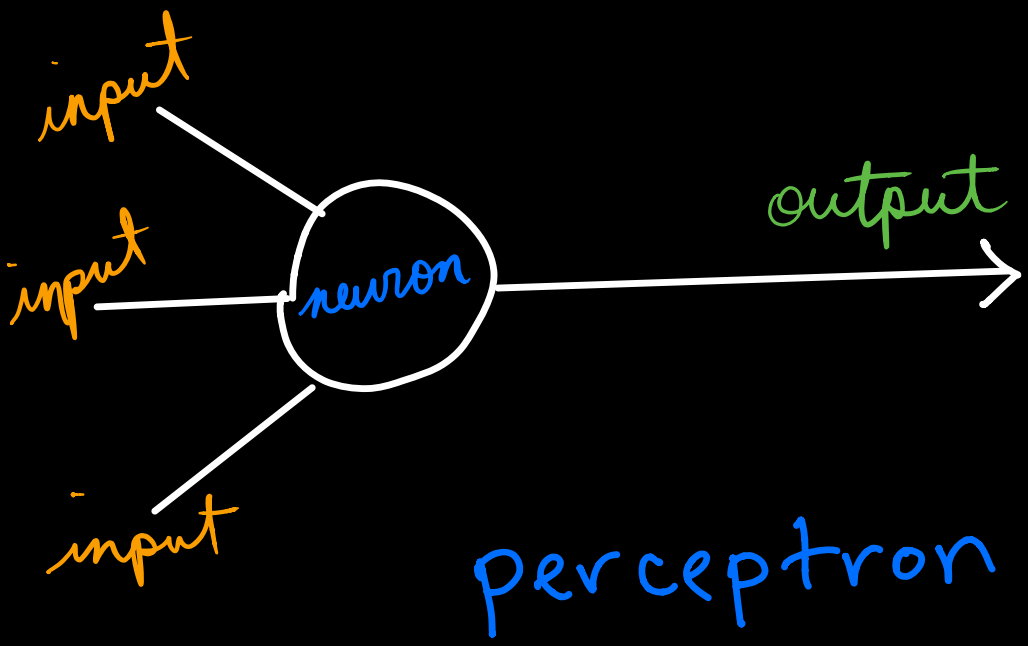
JON KROHN'S
"Fundamentals of Deep Learning"
Reference PDF
(jonkrohn.com)



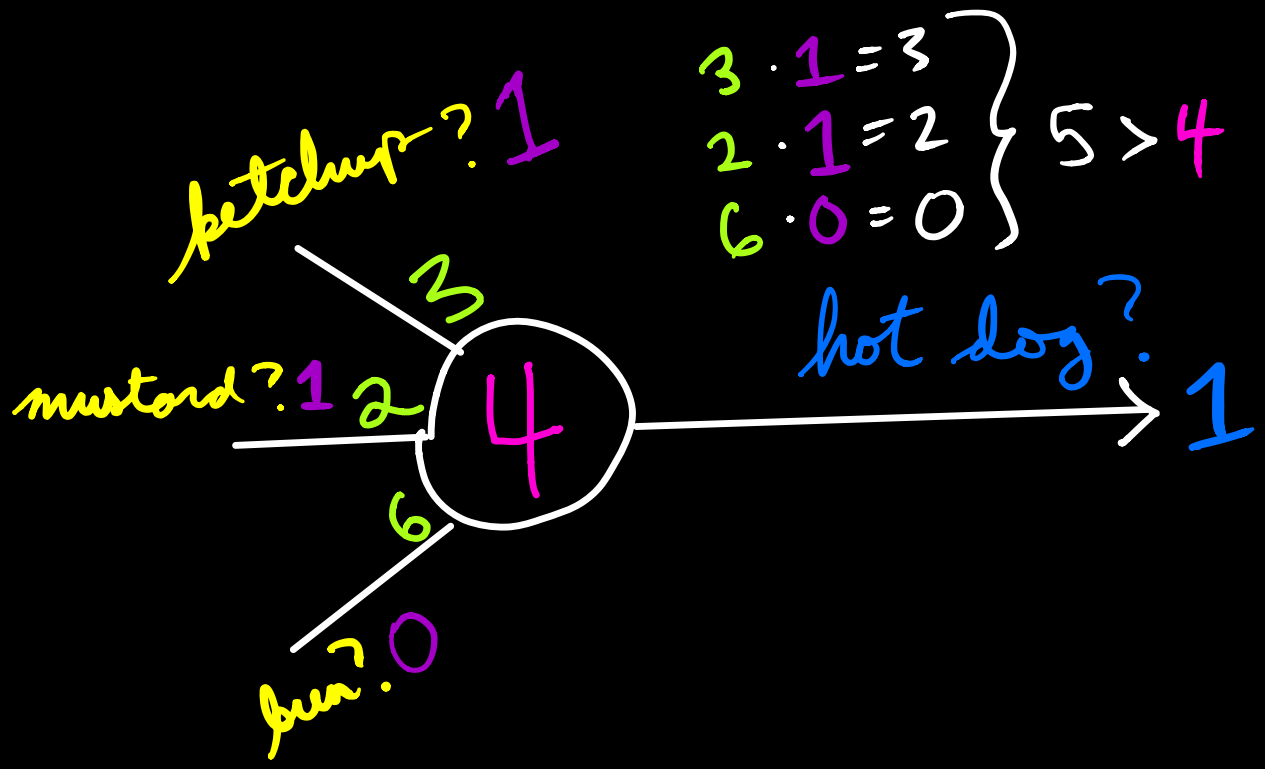


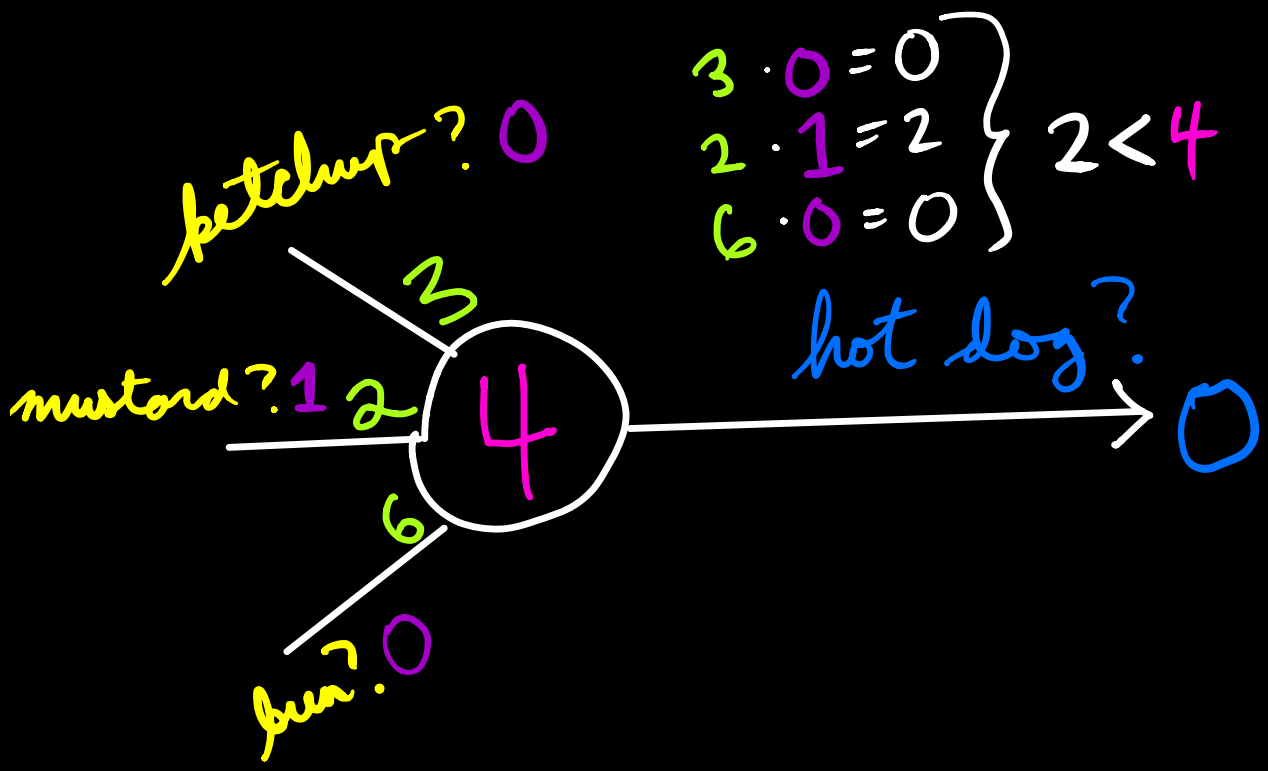
once charge of cell body gets below a value it will fire action potential
dendrites are inputs

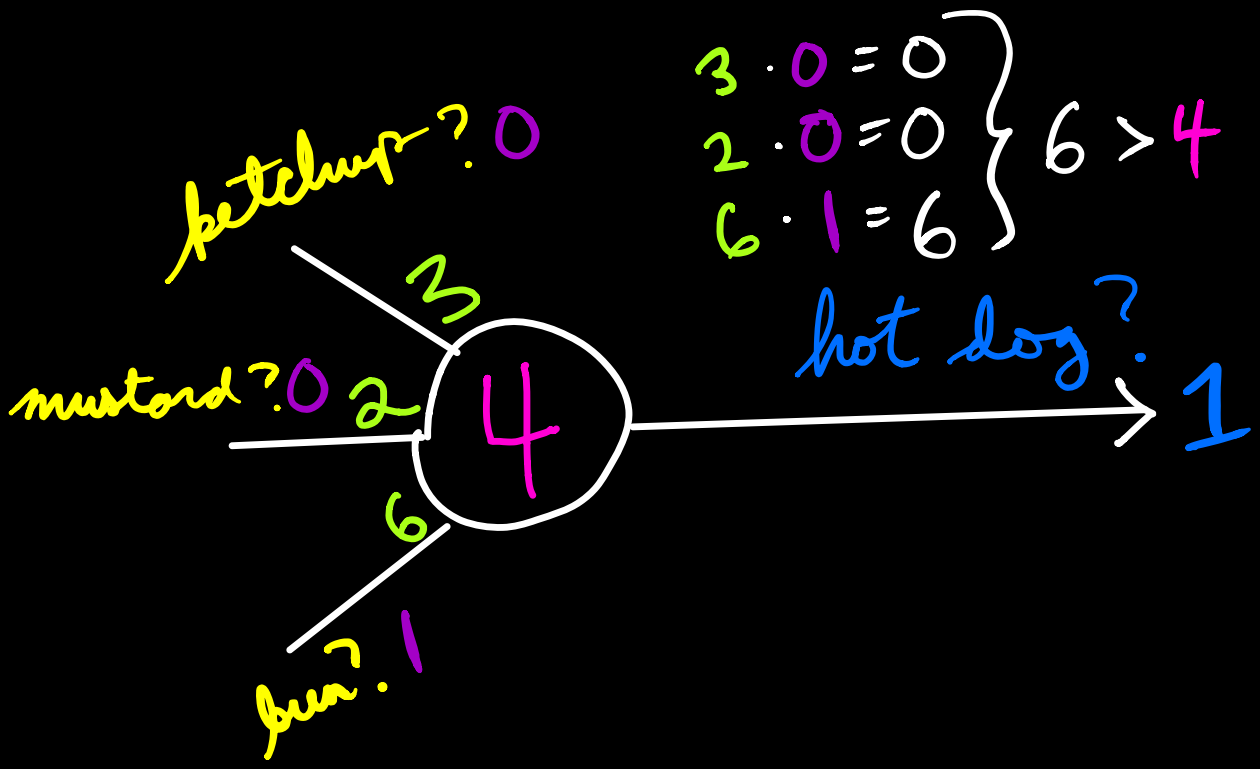




perceptron


ketchup? 1
3
mustard? 1
2
4
6
bun? 0
3 · 1 = 3
2 · 1 = 2
6 · 0 = 0
5 > 4
hot dog?
1


ketchup? 0
3
mustard? 1
2
4
6
bun? 0
3 · 0 = 0
2 · 1 = 2
6 · 0 = 0
2 < 4
hot dog?
0

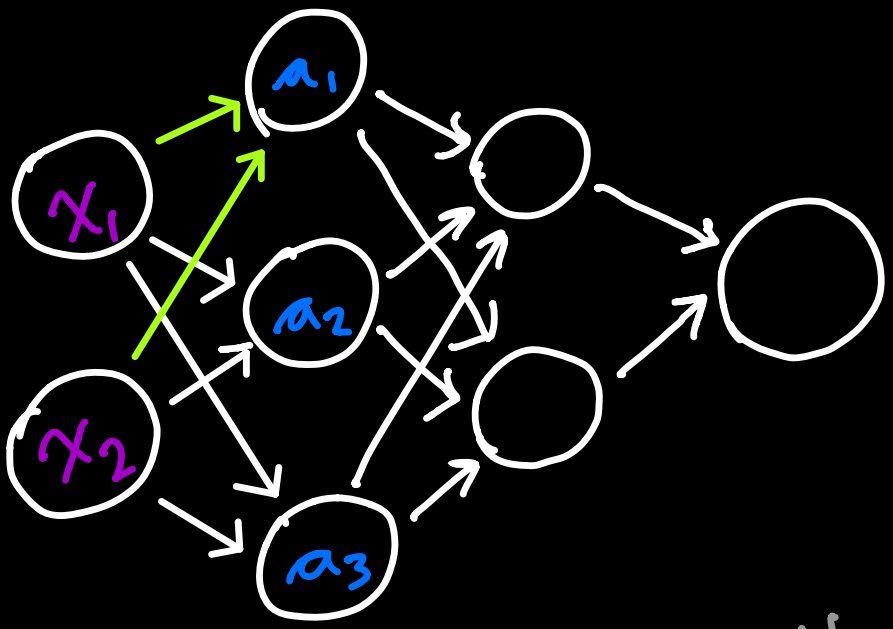


if $\sum_{i=1}^{n} w_i x_i >$ threshold, output 1
otherwise, output 0

$b \equiv -\text{threshold}$

$$\text{output} \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\boxed{w \cdot x + b}$$

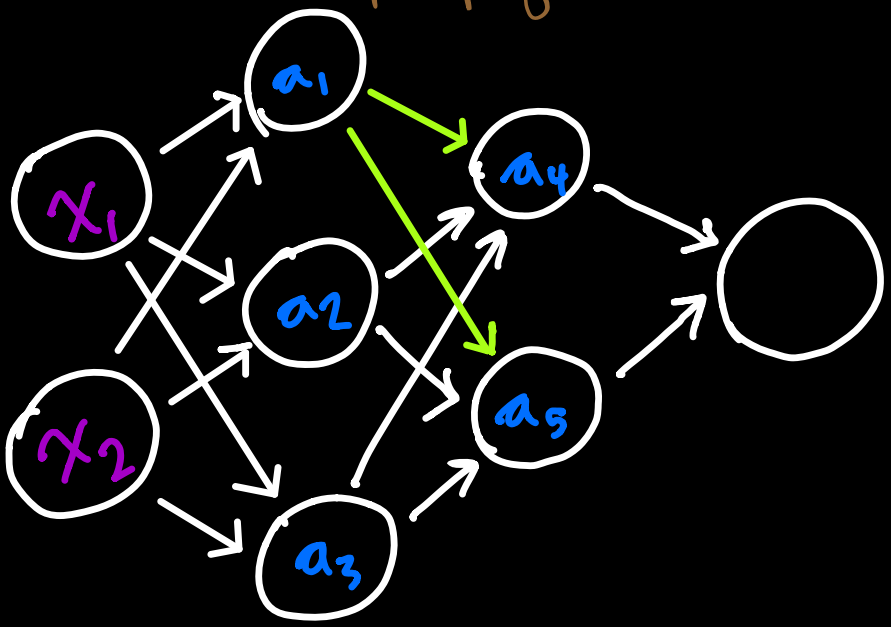


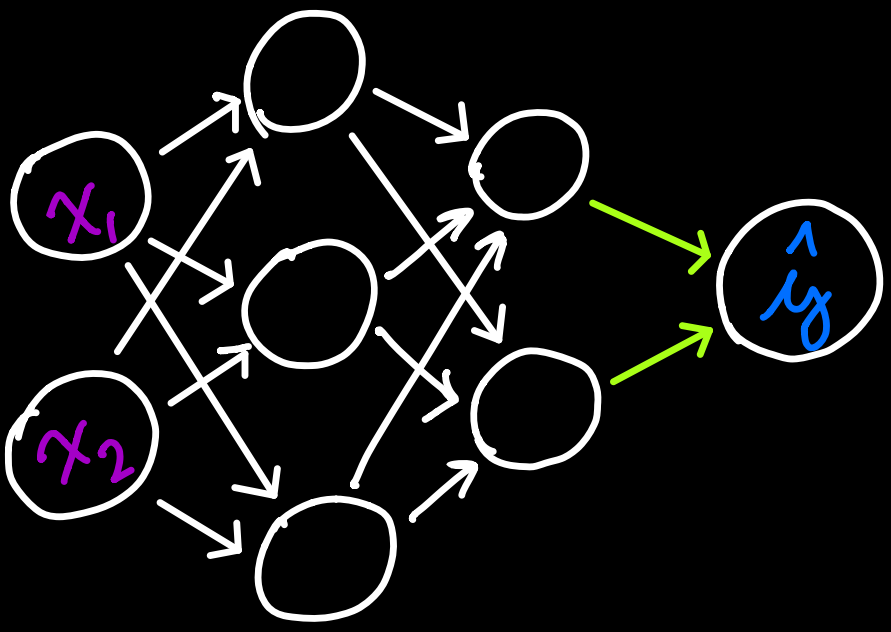
layer 1 2 3 4

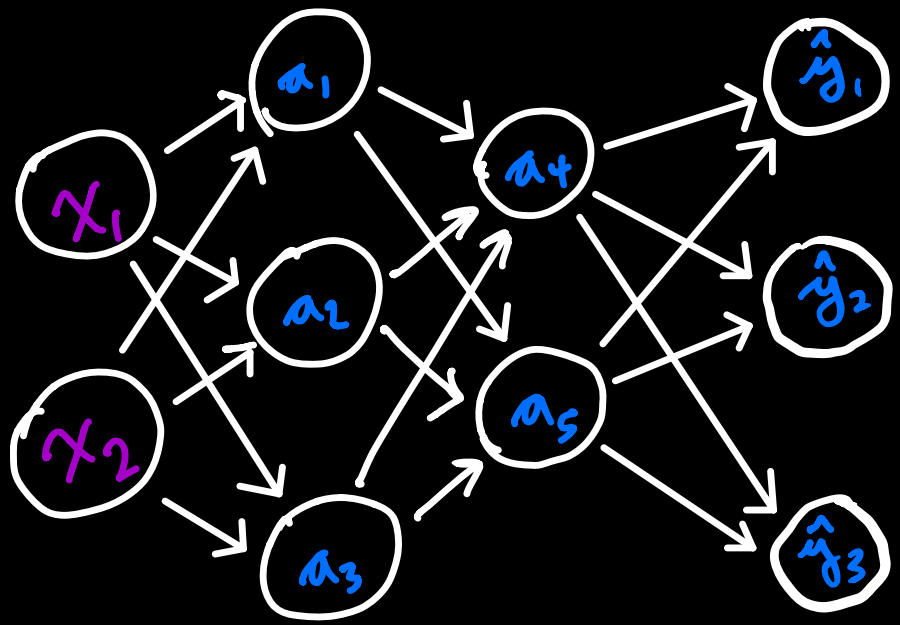
hidden layer 1 2

forward propagation

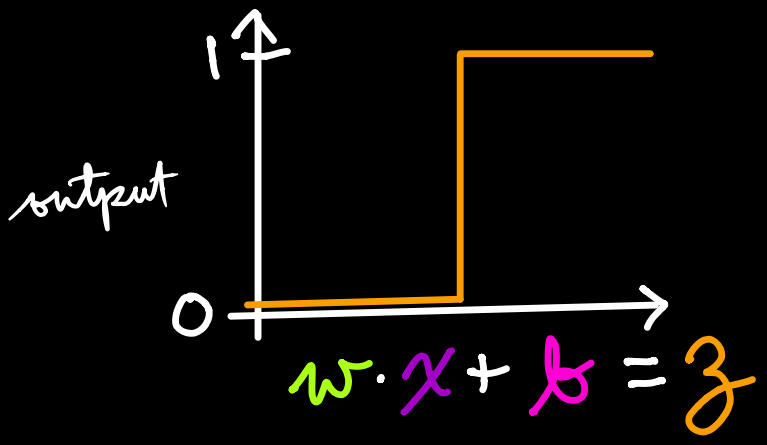
backpropagation

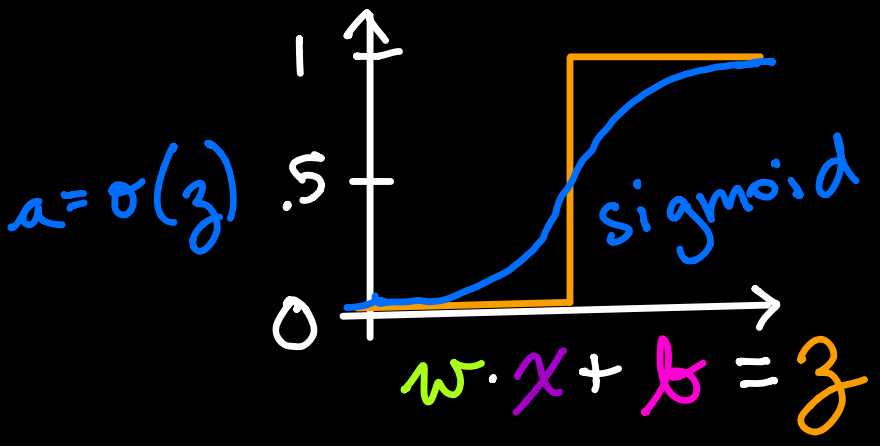



x_1
x_2
$\hat{y}$

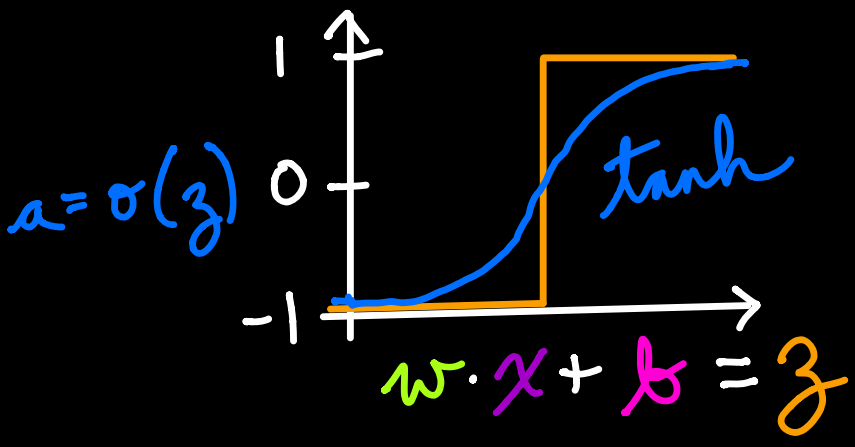

x_1
x_2
a_1
a_2
a_3
a_4
a_5
$\hat{y}_1$
$\hat{y}_2$
$\hat{y}_3$

Activation Function



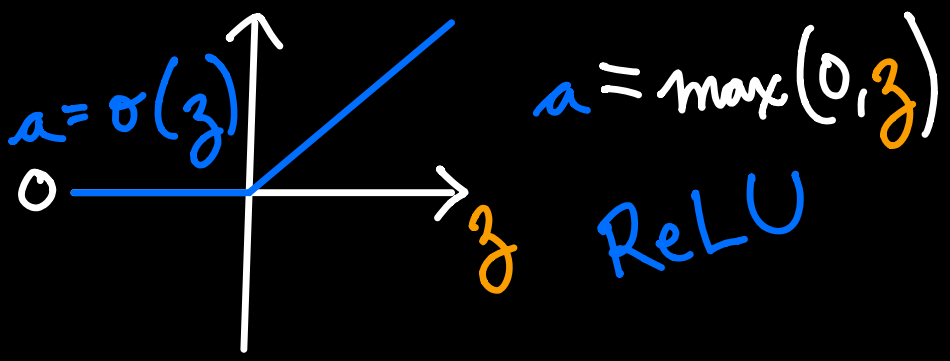


sigmoid you would only use in output layer ie if want to bind between 0 and 1



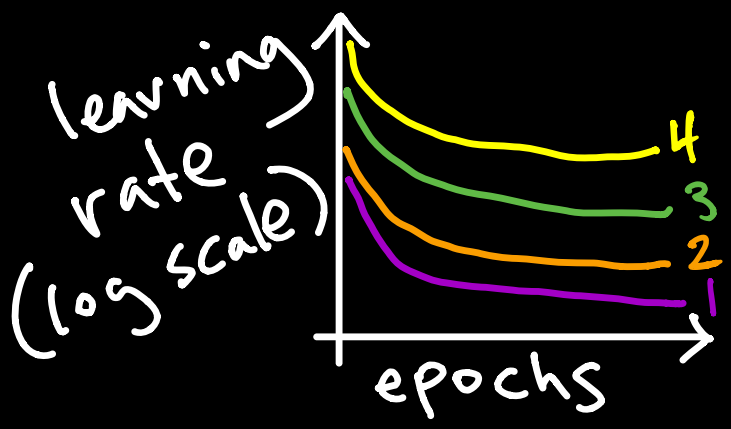
learns much faster than sigmoid because of mean of 0

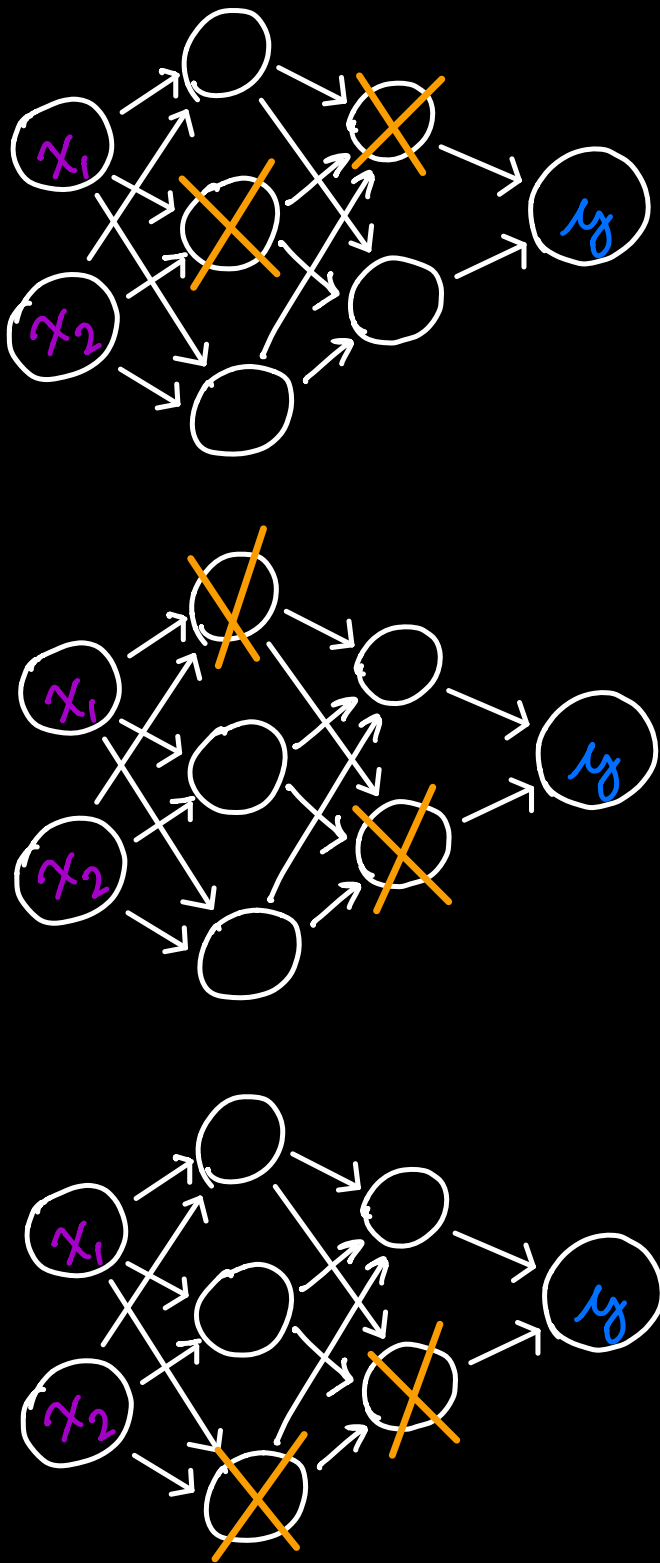
only way to really know tanh or RELU is to try both of them

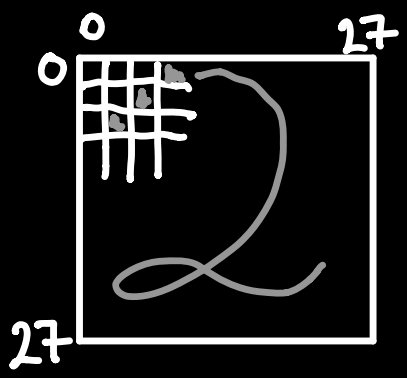


able to learn more quickly on a straight line vs a curve

leaky relu offer small advantage




x_1
x_2
y
x_1
x_2
y
x_1
x_2
y



MNIST

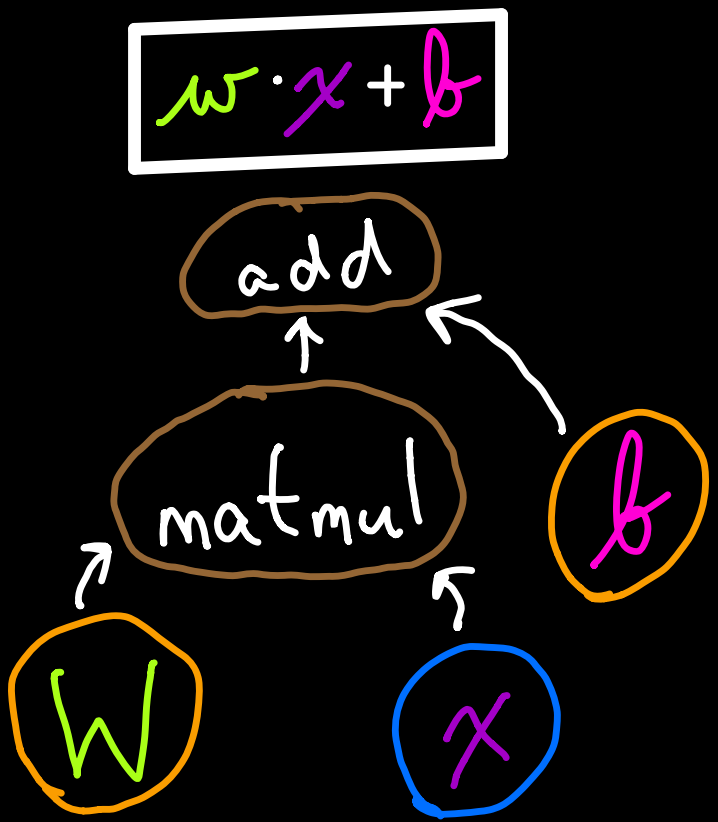
28 x 28 = 784 input

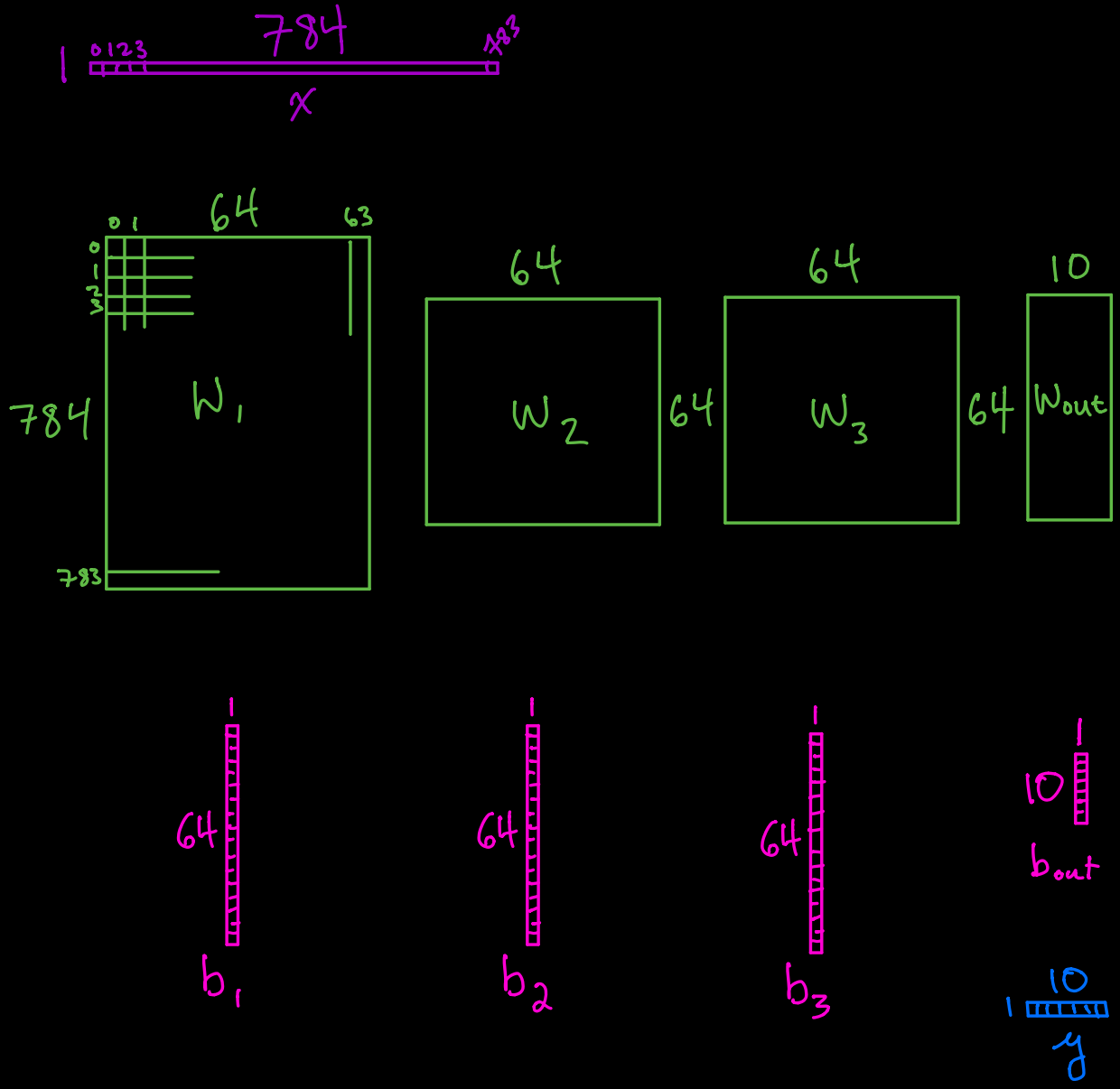
↓

64 sigmoid hidden

↓

10 softmax output




784
0 1 2 3
783
x
64
0 1
63
0
1
2
3
784
W_1
783
64
W_2
64
64
W_3
64
10
W_{out}
64
b_1
64
b_2
64
b_3
10
b_{out}
10
y