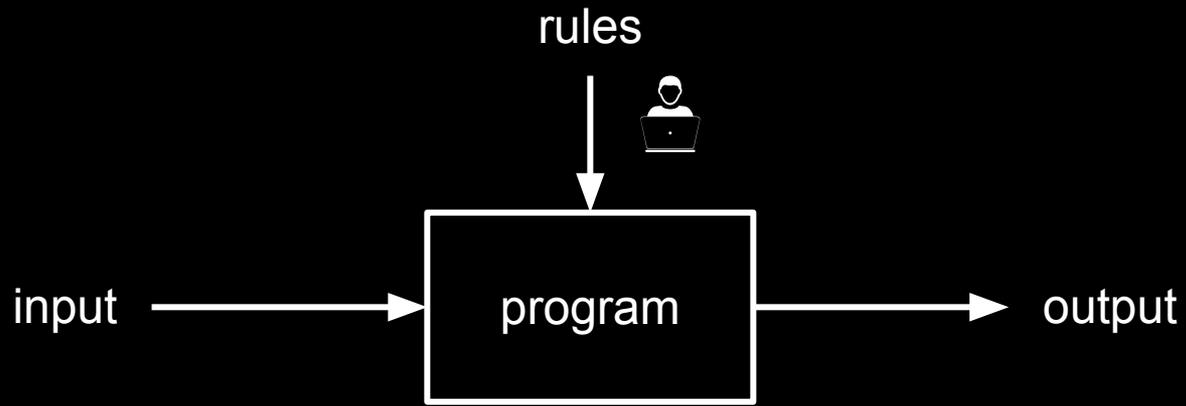
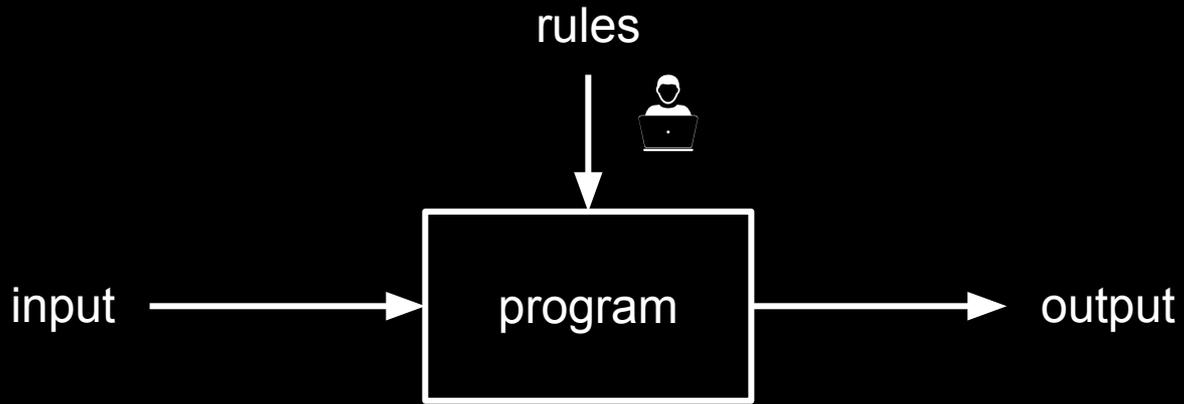
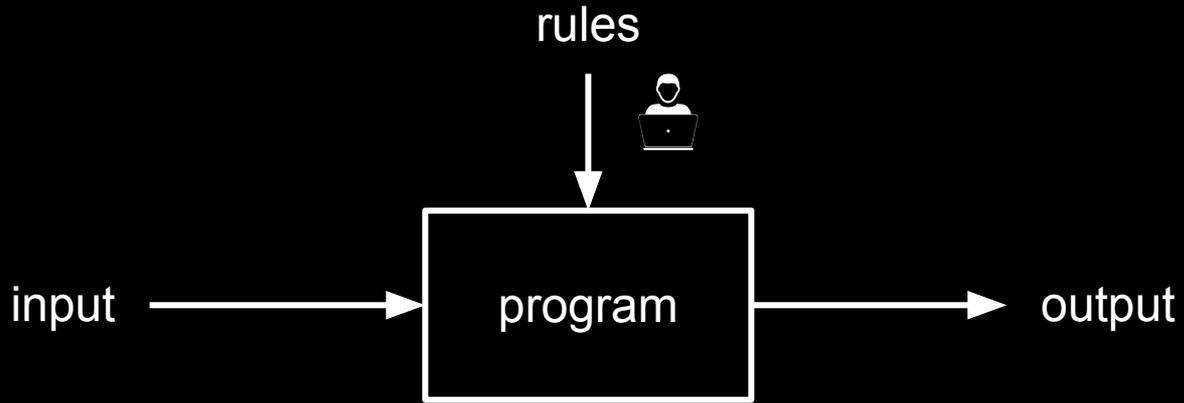


# LEARNING

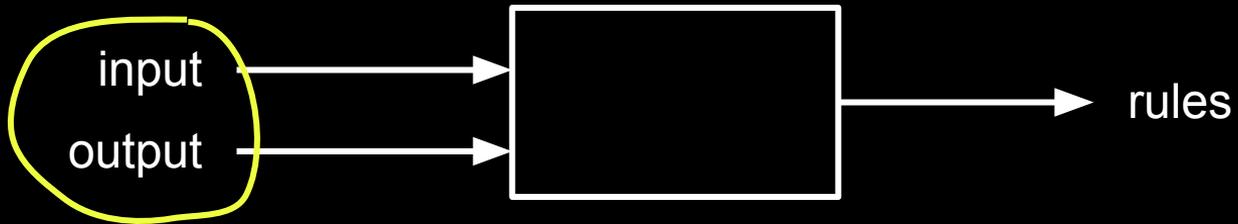


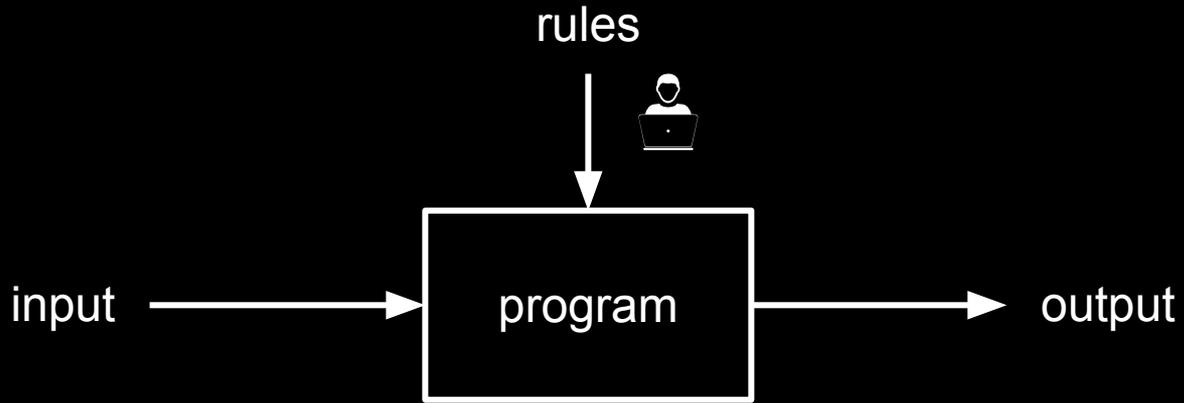


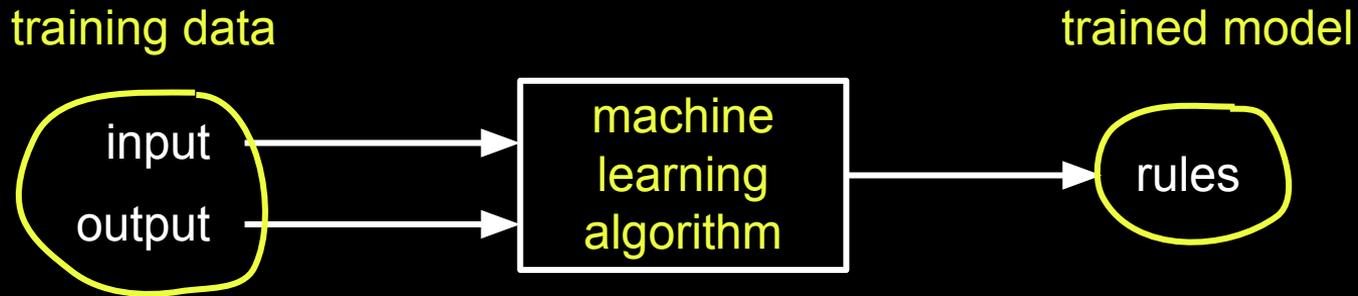
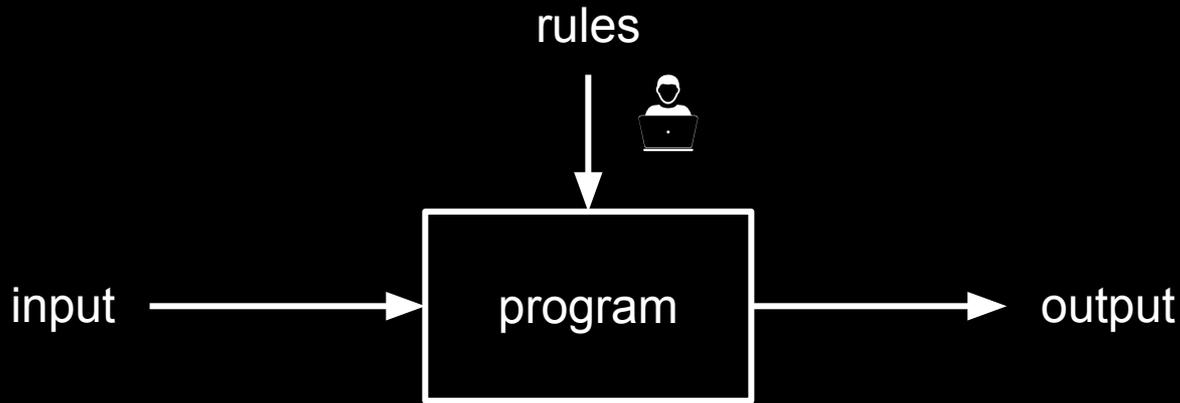


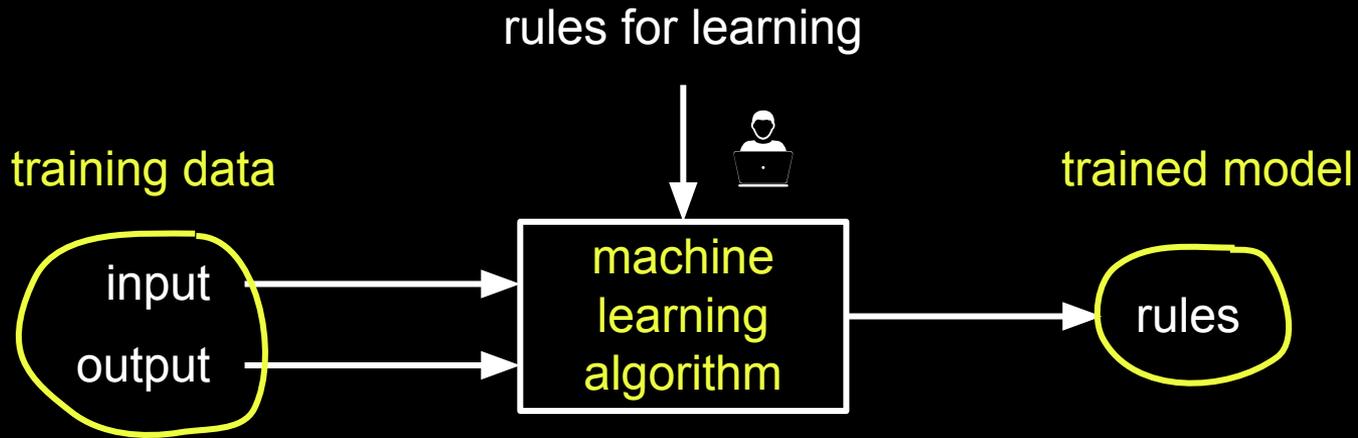
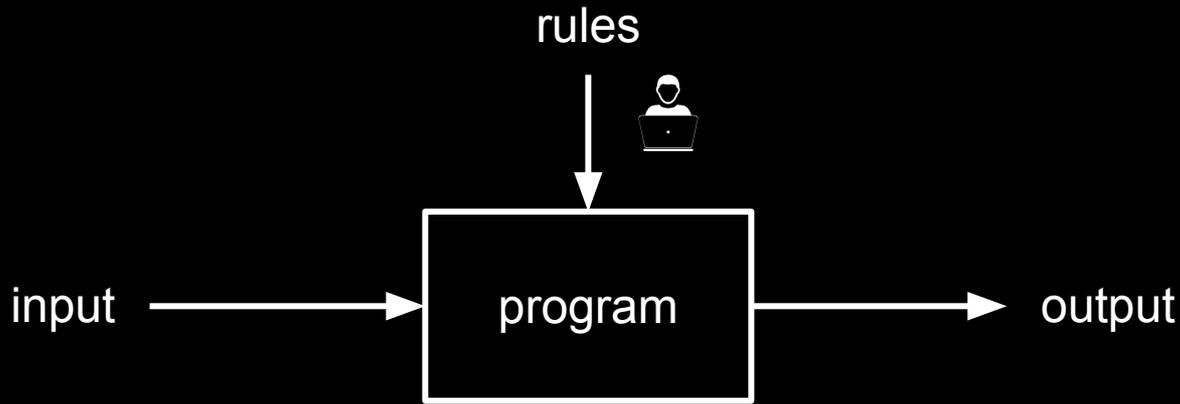


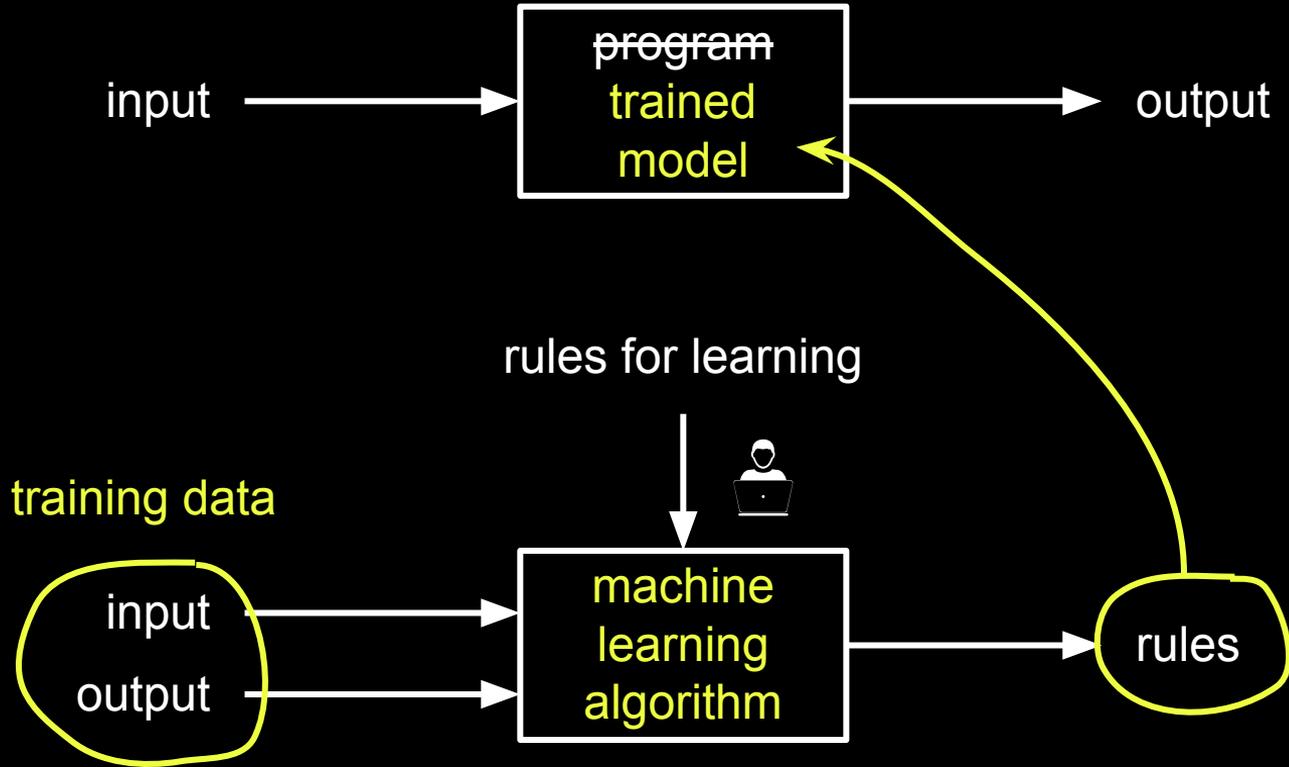
training data

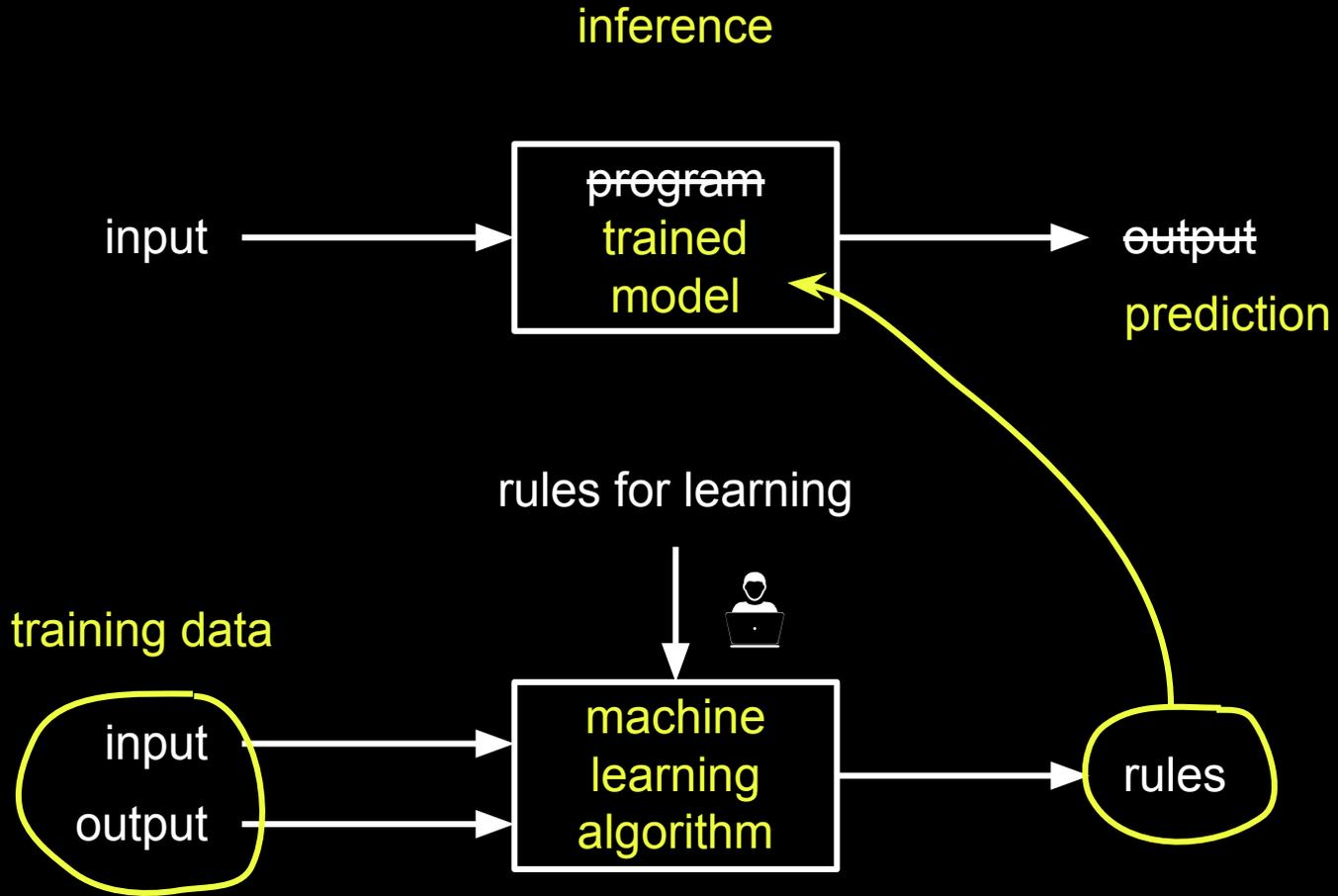














"yes" / "no"

new email



prediction

rules for learning



training data



features



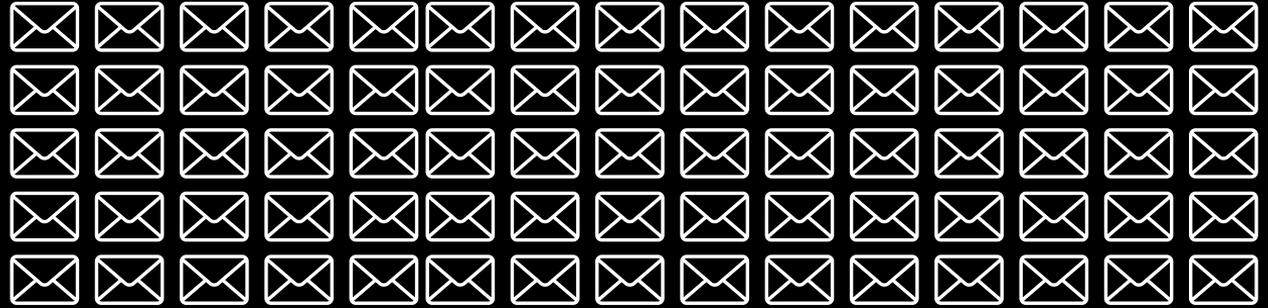
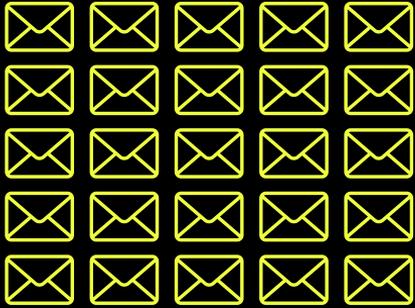
labels



naive bayes classifier

no yes no

$$P(\textit{spam}|\textit{words}) = \frac{P(\textit{words}|\textit{spam}) \times P(\textit{spam})}{P(\textit{words})}$$



spam

no spam

total	25		75	
"buy"	20	$20/25 = 0.8$	5	$5/75 \approx 0.066$
"cheap"	15	$15/25 = 0.6$	10	$10/75 \approx 0.133$
"buy" & "cheap"	12	$0.8 \times 0.6 = 0.48$	$\approx 0.67$	$0.066 \times 0.133 \approx 0.0089$

$$P(\text{spam} \mid \text{buy \& cheap}) = 12 / (12 + 0.67) \approx 0.947$$

- I. we can write rule-based programs to solve some problems
- II. some problems are too complex to be solved with rules alone
- III. for some problems, we cannot define a suitable set of rules to solve it
- IV. for some problems, we can have a computer program learn the rules from data (machine learning)

$x_1$	$x_2$	$y$
4	2	8
1	2	5
0	5	10
2	1	4
3	2	?

$x_1$	$x_2$	$y$
4	2	8
1	2	5
0	5	10
2	1	4
3	2	?

which function produces the  
observed pattern?

$$f(x_1, x_2)$$

in this case, it's easy to spot  
a potential candidate

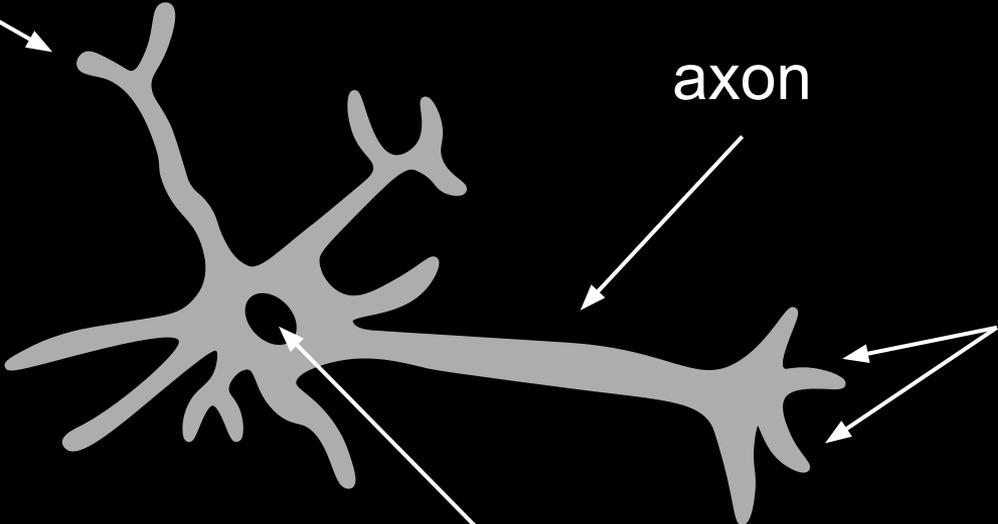
$$y = x_1 + 2x_2$$

$$y = w_1 x_1 + w_2 x_2$$

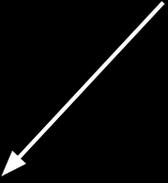
what if it's not so easy?



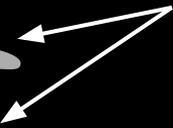
dendrites



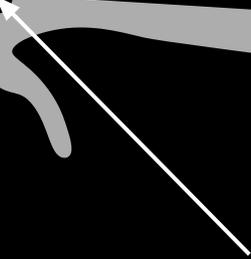
axon

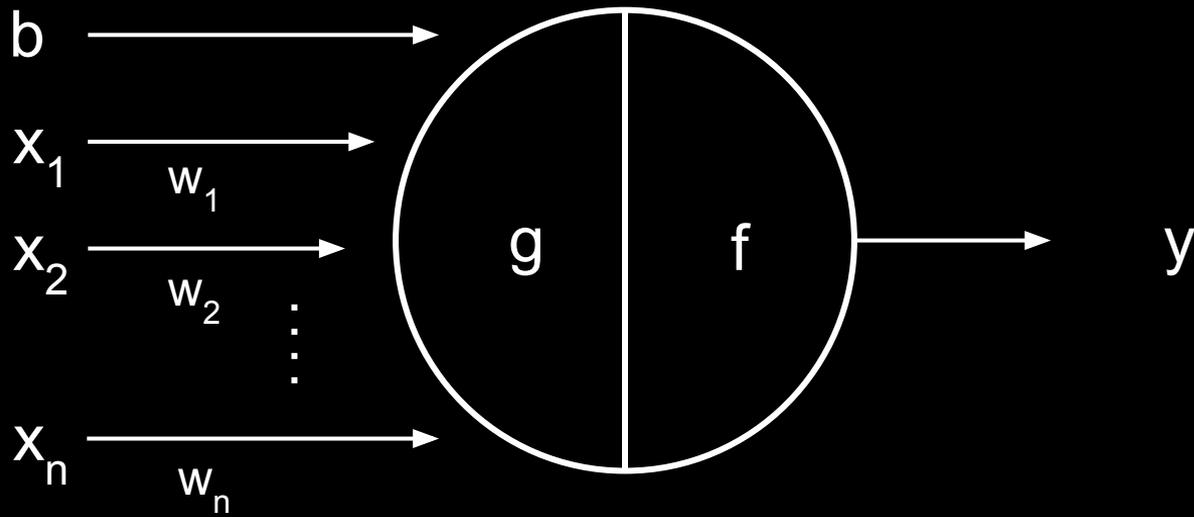


axon terminals



cell body





perceptron

multi-layered  
perceptron

backpropagation

is my password safe?