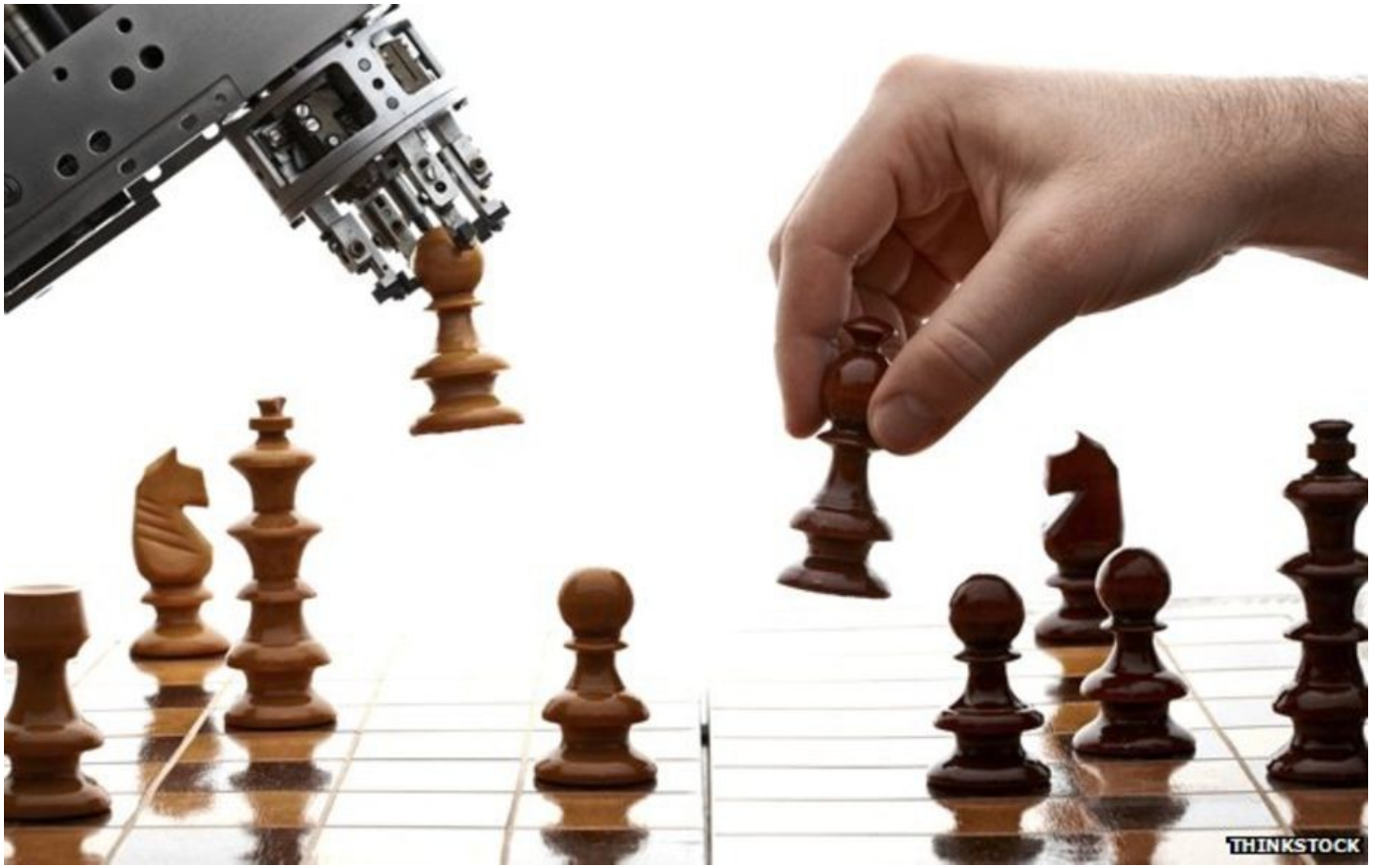


Introduction to Artificial Intelligence

COSC 4550 / COSC 5550

Professor Cheney

10/18/17

when in doubt, choose the simplest model possible

make sure the the number of data points you have is not less than the number of variables in your model (underdetermined system)
and aim for at least an order of magnitude more data

removing features (attributes) from you dataset is a similar way to simplify optimization

creates a smaller dimensional problem and can lead to less overfitting

a good place to start is
removing attributes with low variance
(if they don't change much, they probably can't explain
the changes in your other variables much either)

removing correlated attributes can also be effective for
simplifying you model with minimal predictive loss
(if two features are both describing the same thing,
the second probably isn't telling you anything the first isn't)

(more in this later)

regularization

explicitly penalize your model for using features

make sure it's only paying attention
to the ones that really matter!
(those useful enough to make up
for the penalty of using them)

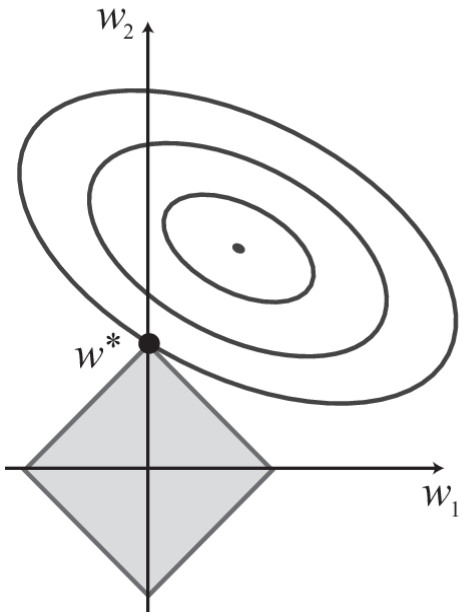
$$\text{Cost}(h) = \text{Loss}(h) + \text{Complexity}(h)$$

how do you measure complexity of a model?

L1-regularization	L2-regularization
Complexity = $\Sigma_{i:n}\|w_i\|$ (sum of magnitude of weights)	Complexity = $\Sigma_{i:n}\|w_i\|^2$ (sum of magnitude of weights squared)
minimize total # of parameters used	minimizes total combined magnitude of parameters

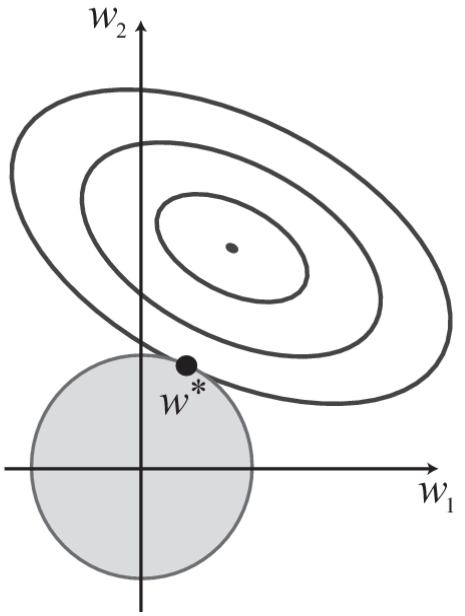
how do you measure complexity of a model?

L1-regularization



minimize total # of parameters used

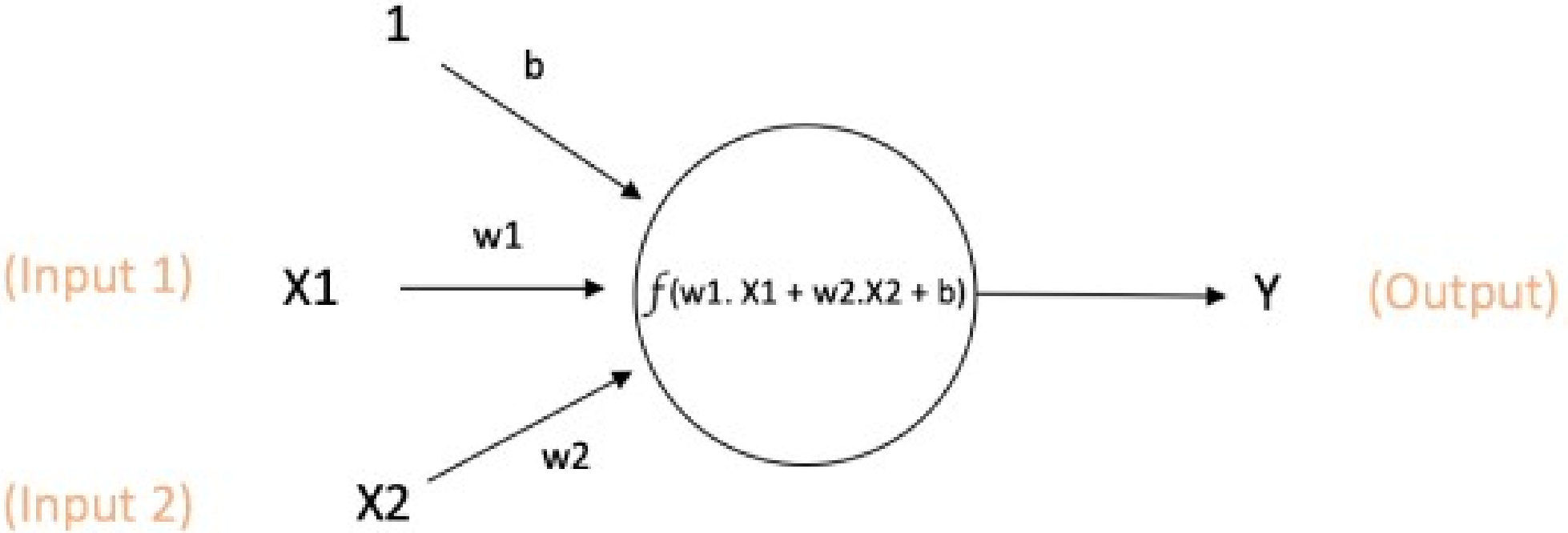
L2-regularization



minimizes total combined magnitude of parameters

neural networks

artificial neuron



Output of neuron = Y= f(w1. X1 + w2.X2 + b)

if “activation function” f(x) is:

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

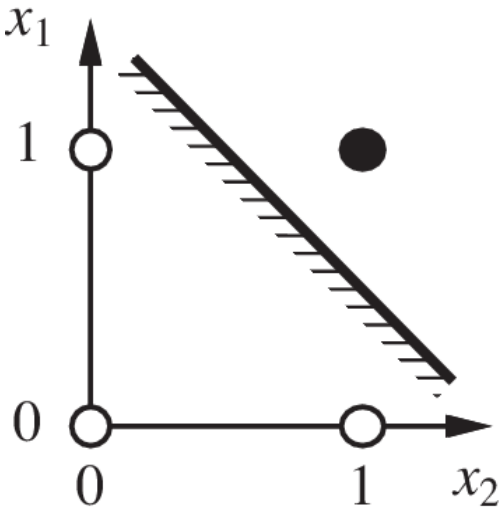
called a “perceptron”

this basic framework is not new…

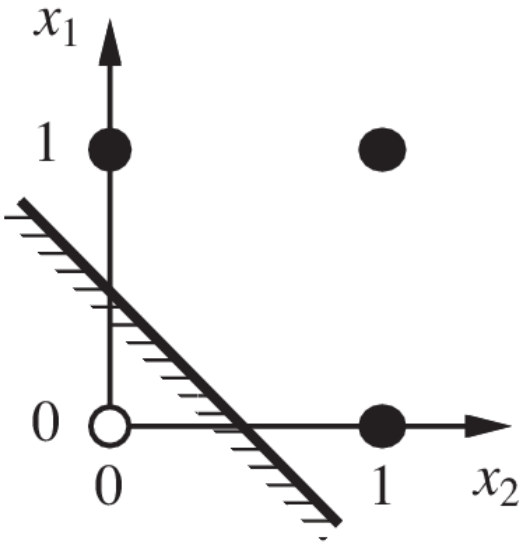
first artificial neuron:
McCulloch & Pitts (1943)

perceptron:
Rosenblatt (1958)

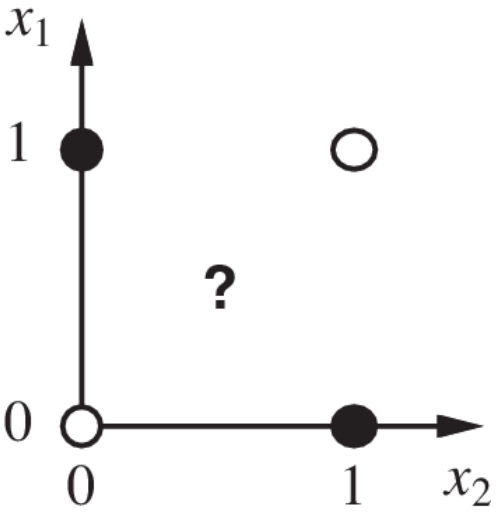
perceptron as logical opperator



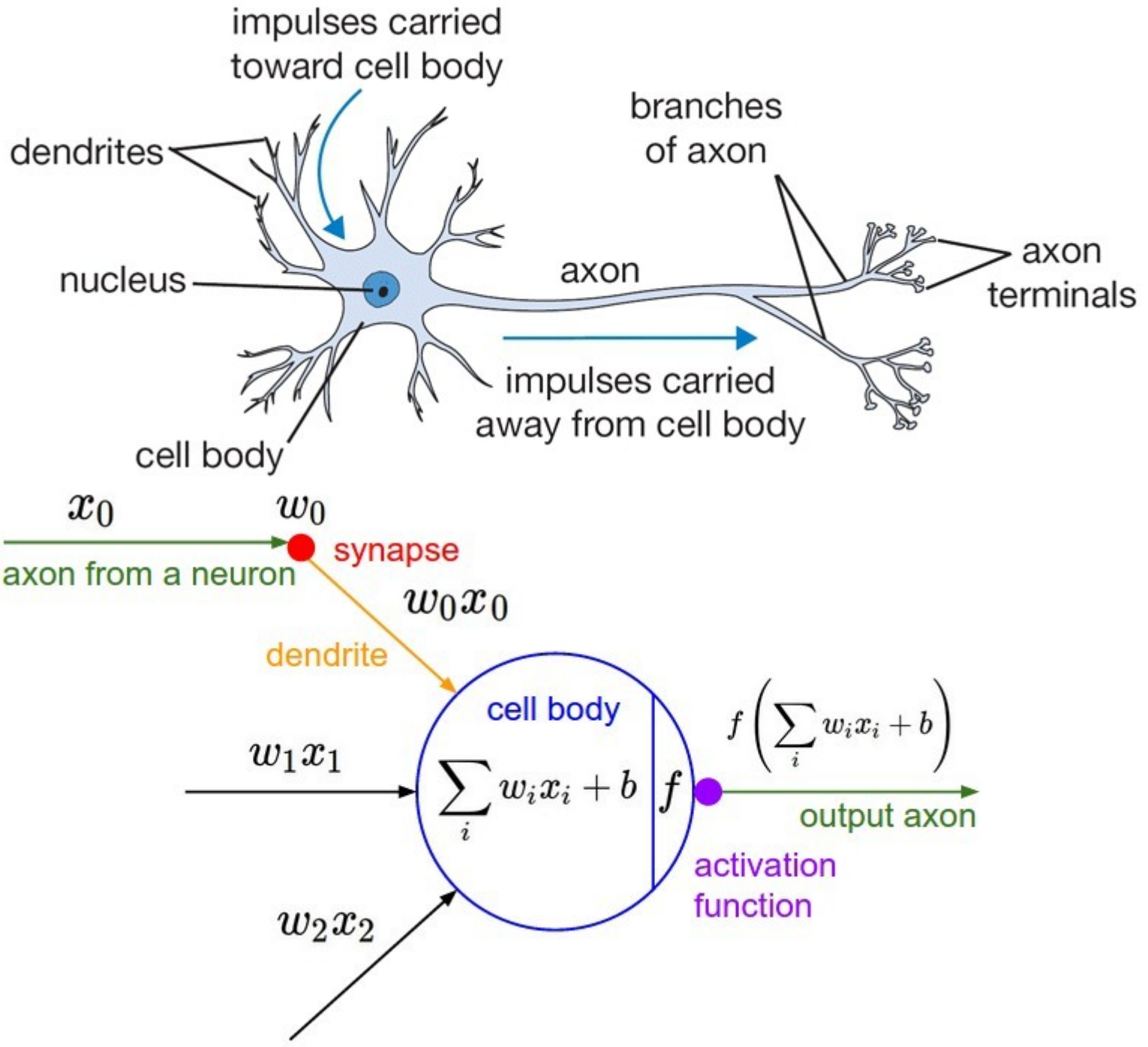
(a) x_1 **and** x_2



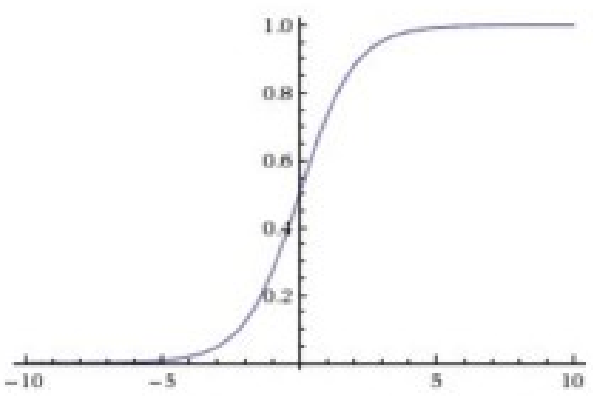
(b) x_1 **or** x_2



(c) x_1 **xor** x_2


impulses carried
toward cell body
dendrites
branches
of axon
nucleus
axon
axon
terminals
impulses carried
away from cell body
cell body
x_0
w_0
synapse
axon from a neuron
$w_0 x_0$
dendrite
cell body
$f\left(\sum_i w_i x_i + b\right)$
$w_1 x_1$
$\sum_i w_i x_i + b$
f
output axon
activation
function
$w_2 x_2$

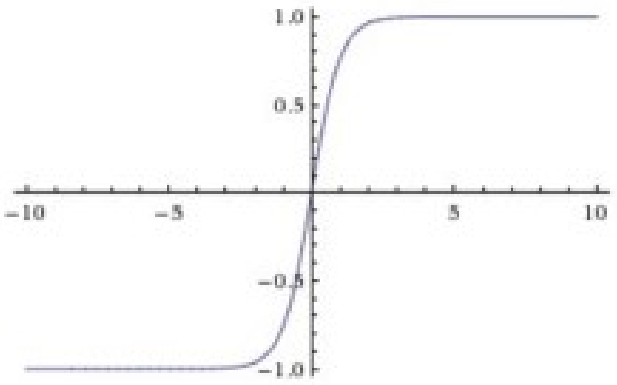
activation functions



Sigmoid

input domain:
[-∞, ∞]

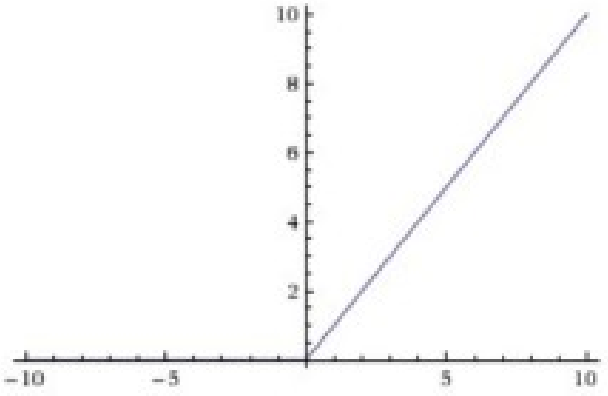
output range:
[0, 1]



tanh

input domain:
[-∞, ∞]

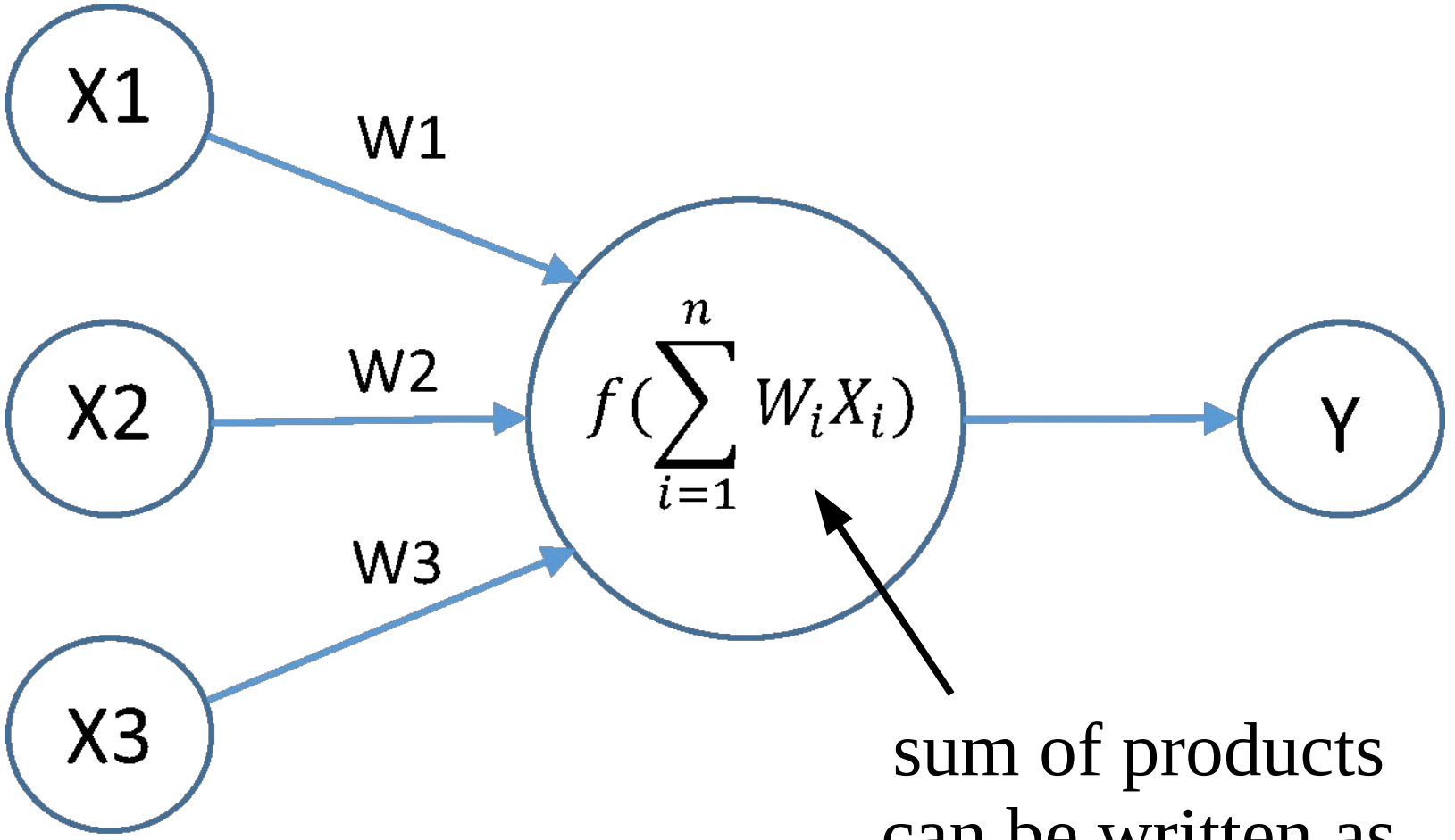
output range:
[-1, 1]



ReLU

input domain:
[-∞, ∞]

output range:
[0, ∞]



sum of products can be written as the dot product of vectors! ($w \circ x$)