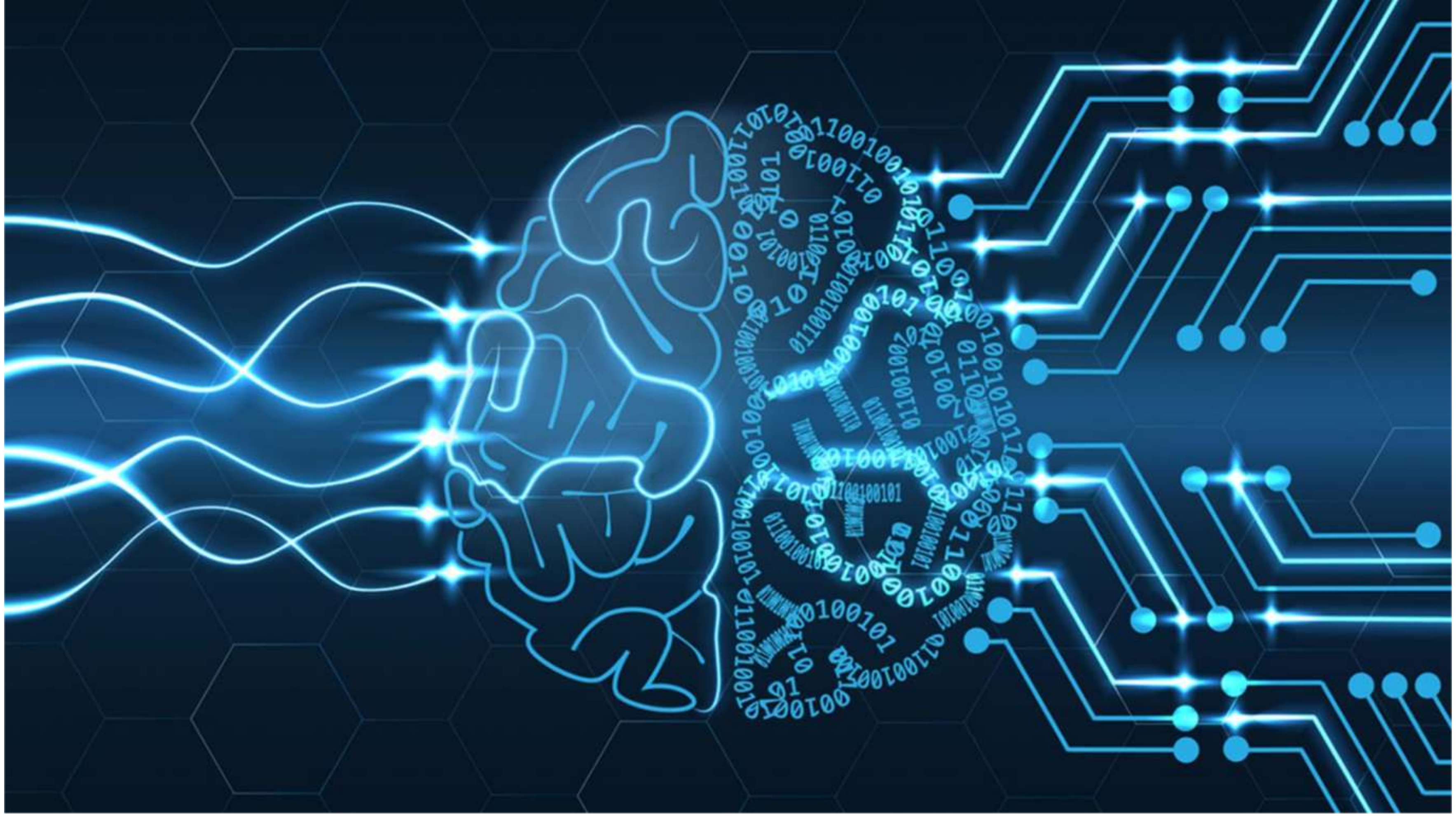
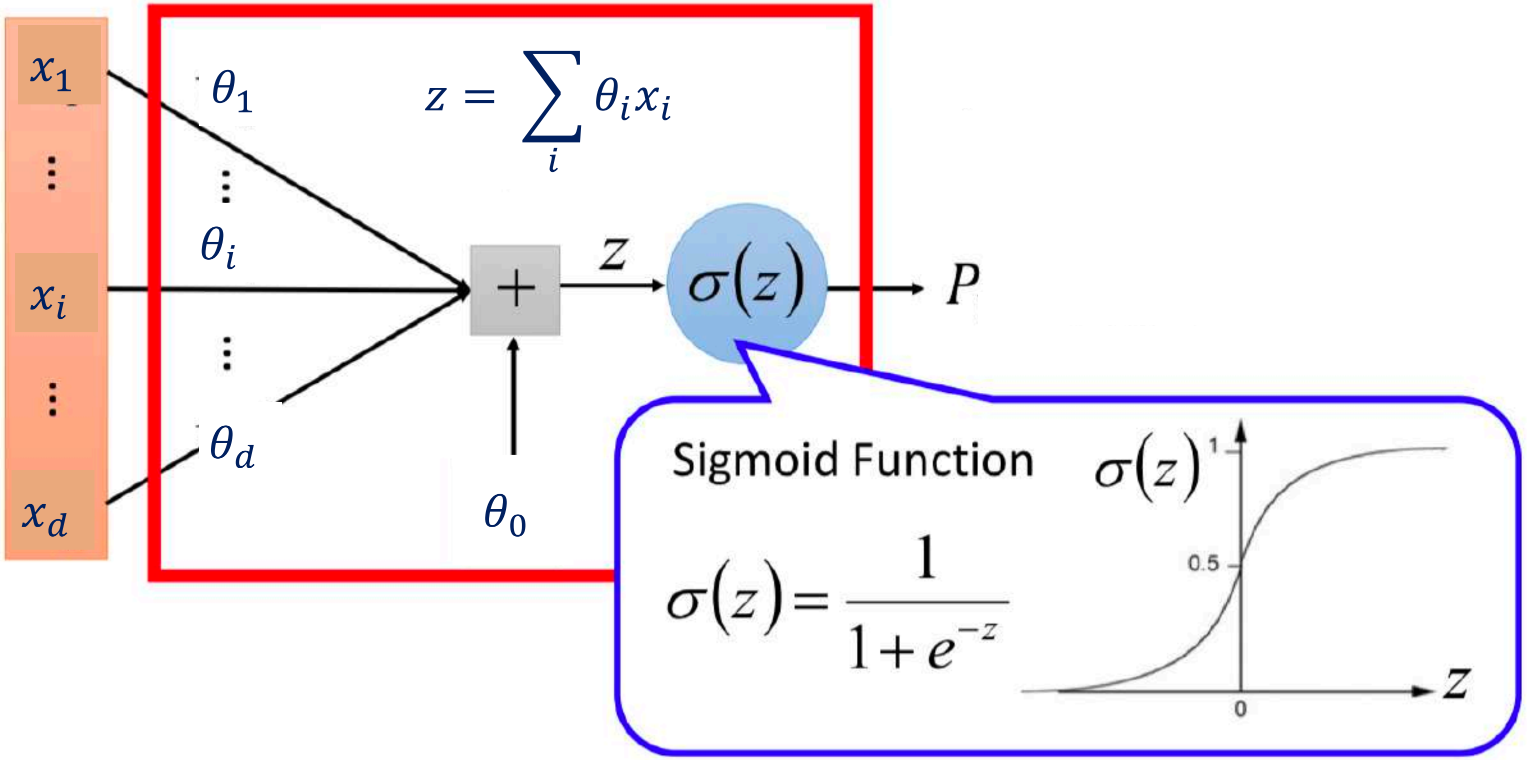


INTRODUCTION TO NEURAL NETWORK

Narges Norouzi

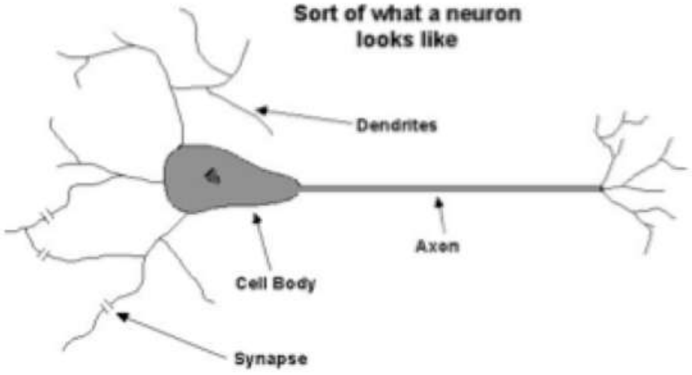
LOGISTIC FUNCTION

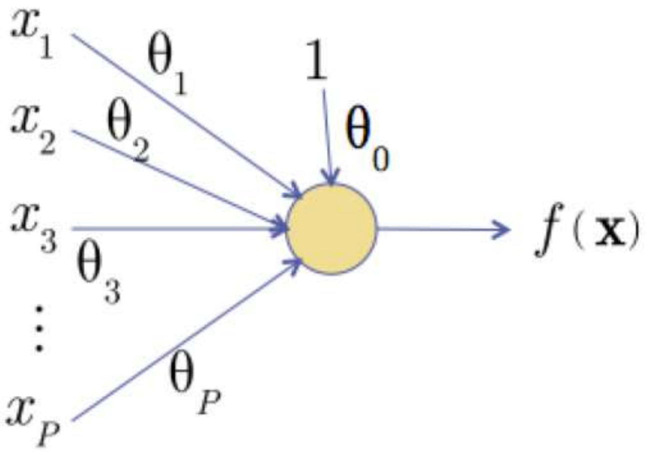


https://walkccc.github.io/CS/ML/5/

THE NEURON METAPHOR

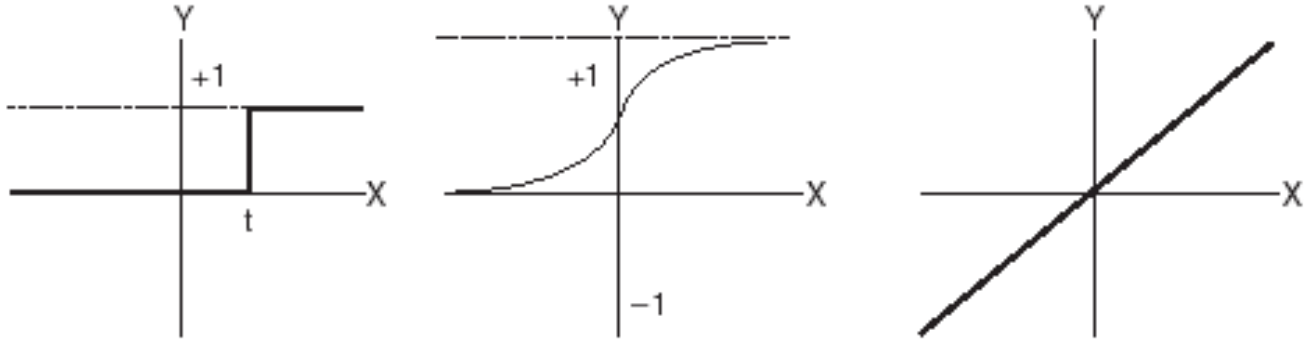
- Neurons
  - Accept information from multiple inputs,
  - Transmit information to other neurons.
- Multiply inputs by weights/parameters along edges
- Apply some function to the set of inputs at each node





ARTIFICIAL NEURONS

- Each neuron in the network receives one or more inputs.
- An activation function is applied to the inputs, which determines the output of the neuron – the activation level.

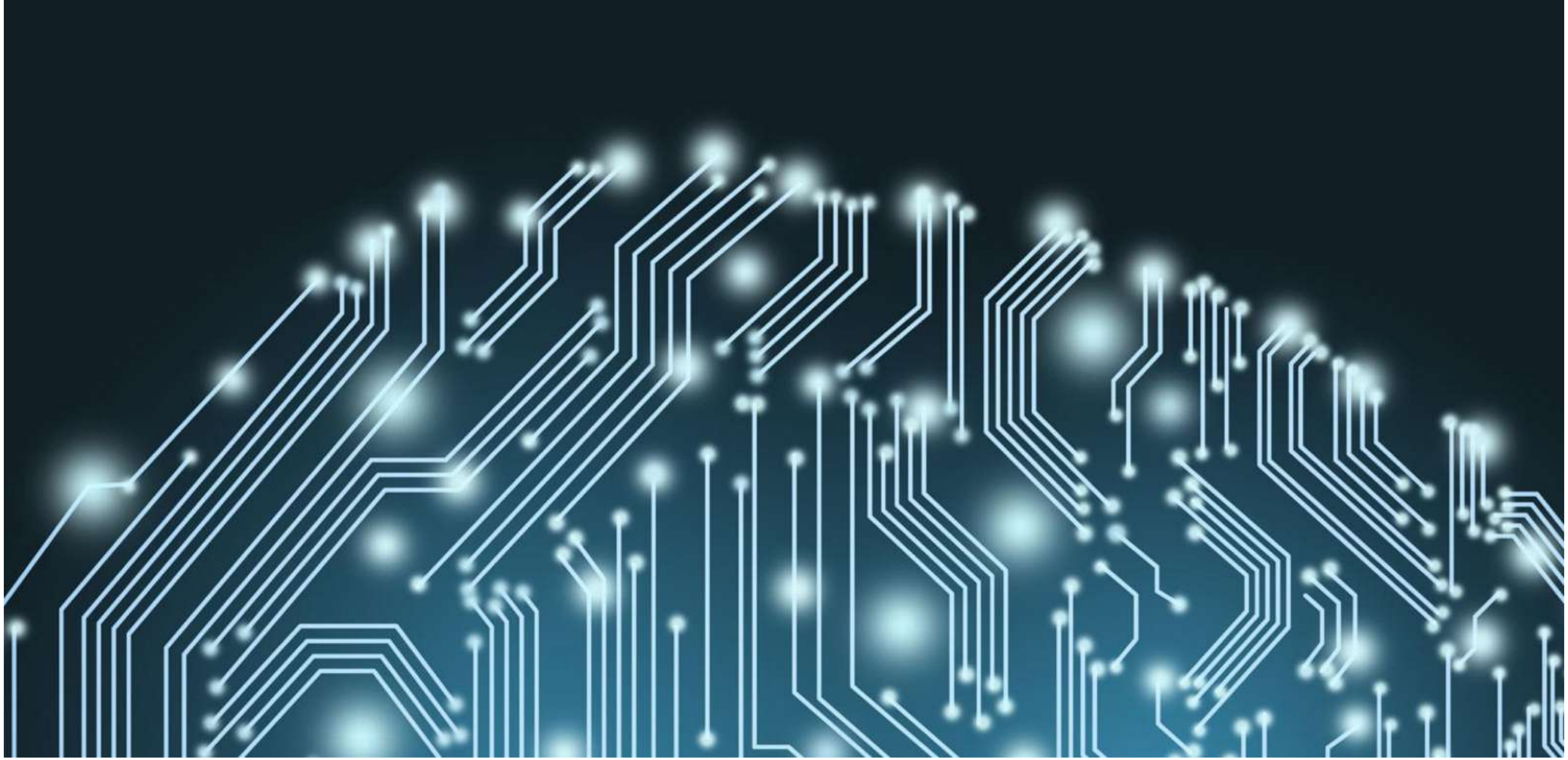


(a) Step funtion (b) Sigmoid funtion (c) Linear funtion

STANDARD ACTIVATION FUNCTIONS

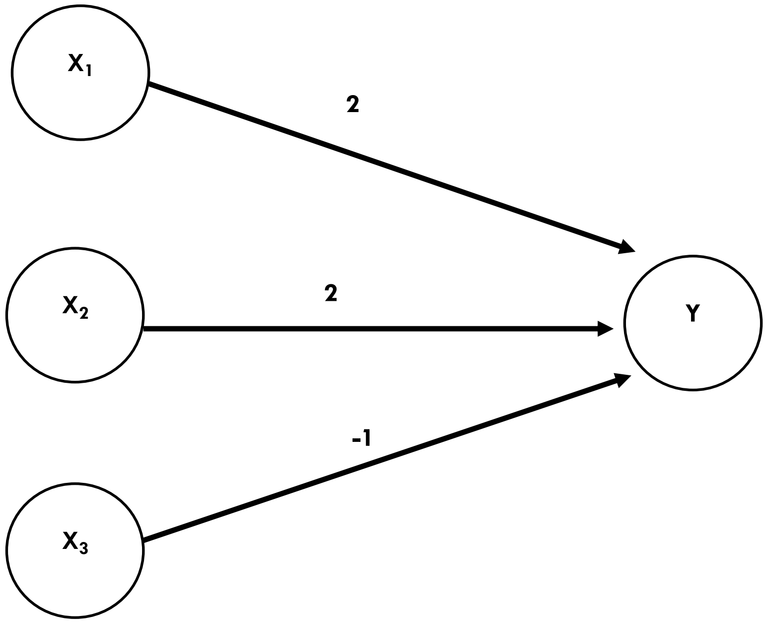
- The hard-limiting threshold function
  - Corresponds to the biological paradigm
    - either fires or not
- Sigmoid function or the logistic function
  - The hyperbolic tangent (symmetrical)
  - Both functions have a simple differential

$$\phi(x) = \frac{1}{1 + e^{-ax}}$$



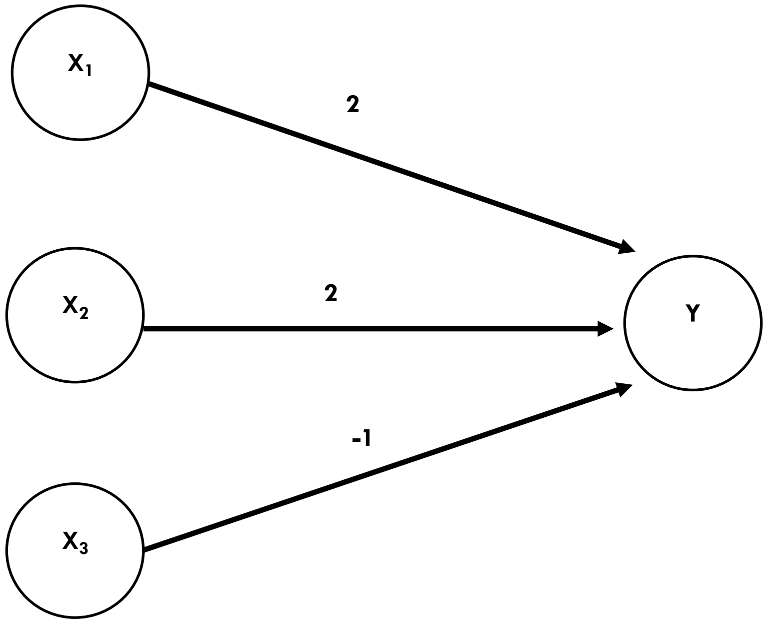
STEP FUNCTION ACTIVATION

STEP FUNCTION PROPERTIES



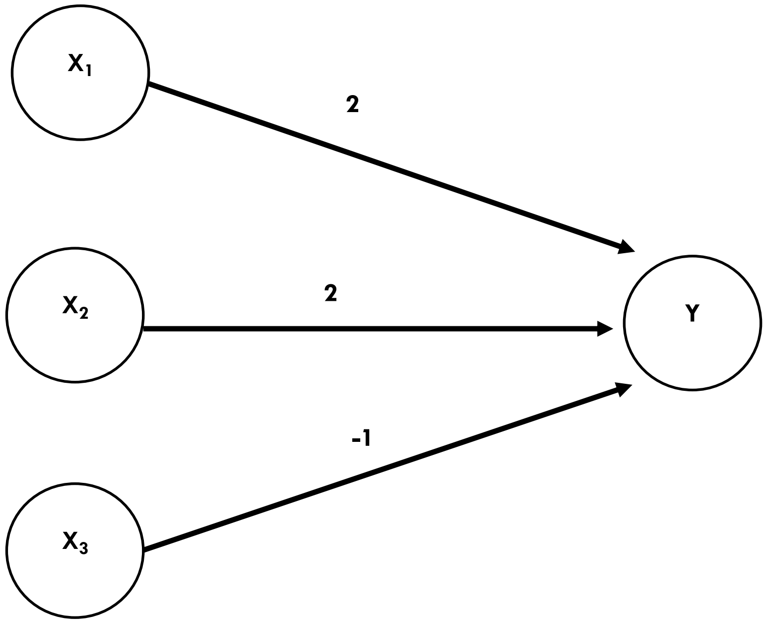
The activation of a neuron is binary. That is, the neuron either fires (activation of one) or does not fire (activation of zero).

STEP FUNCTION PROPERTIES



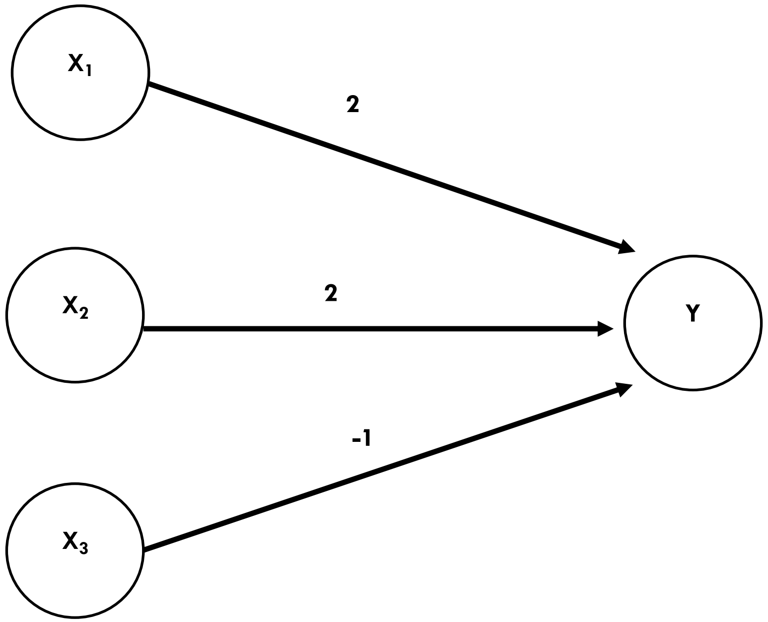
If the weight on a path is positive the path is excitatory, otherwise it is inhibitory.

STEP FUNCTION PROPERTIES



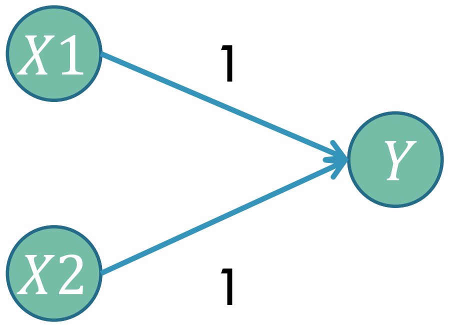
Each neuron has a fixed threshold. If the net input into the neuron is greater than the threshold, the neuron fires. The threshold is set such that any non-zero inhibitory input will prevent the neuron from firing.

STEP FUNCTION PROPERTIES



It takes one time step for a signal to pass over one connection.

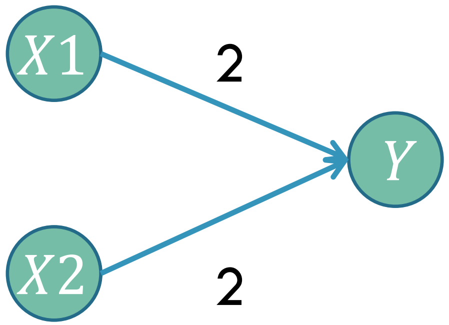
EXAMPLE - AND



AND		
X1	X2	Y
1	1	1
1	0	0
0	1	0
0	0	0

$AND\ Gate\ using\ step\ function\ with$
$Threshold\ (Y)\ =\ 2$

EXAMPLE - OR



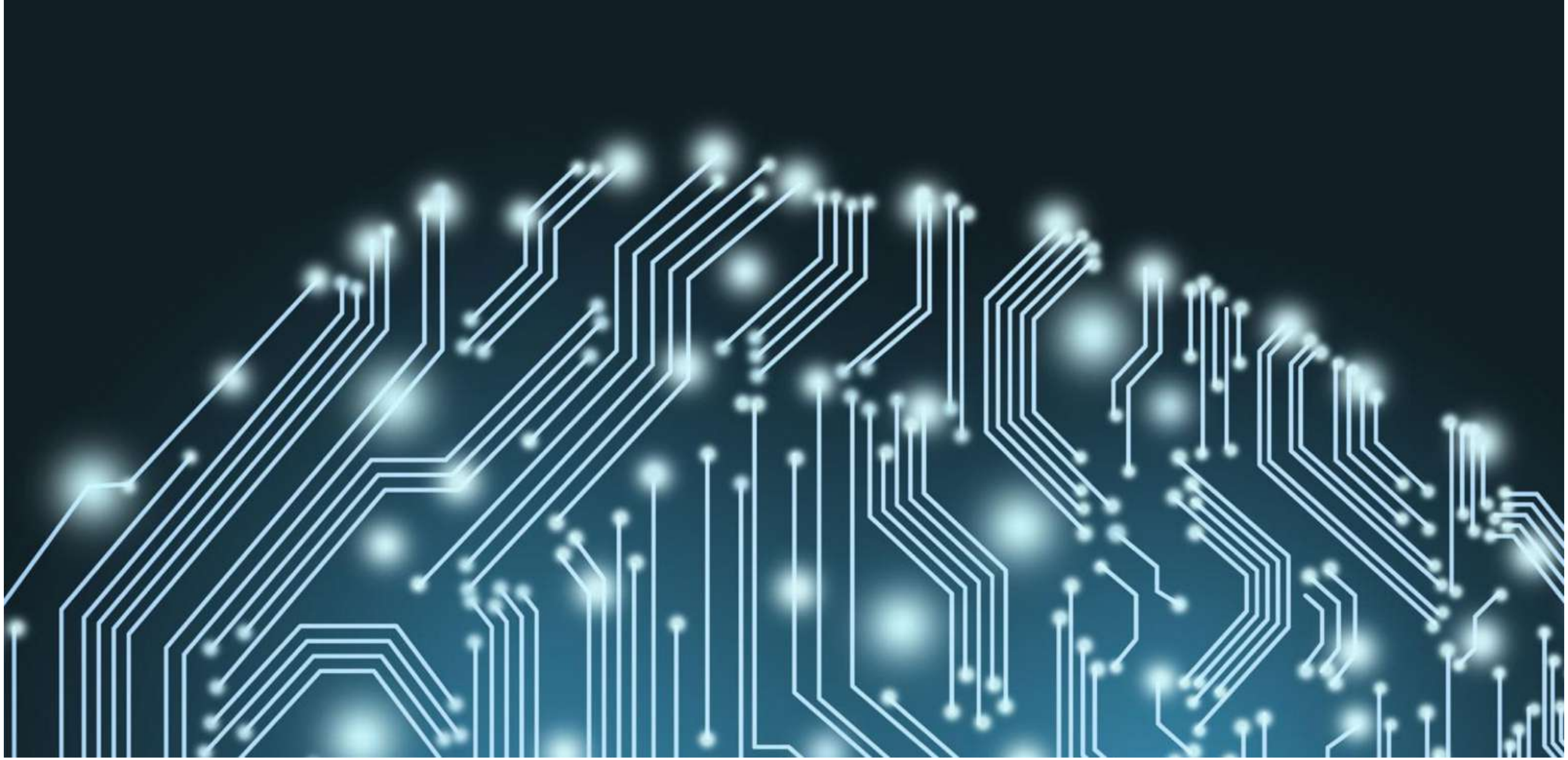
$OR\ Gate\ using\ step\ function\ with$
$Threshold\ (Y)\ =\ 2$

OR		
X1	X2	Y
1	1	1
1	0	1
0	1	1
0	0	0

EXAMPLE - XOR

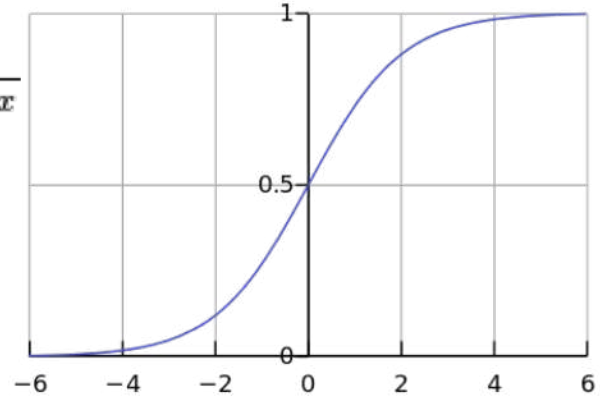
XOR		
X1	X2	Y
1	1	0
1	0	1
0	1	1
0	0	0

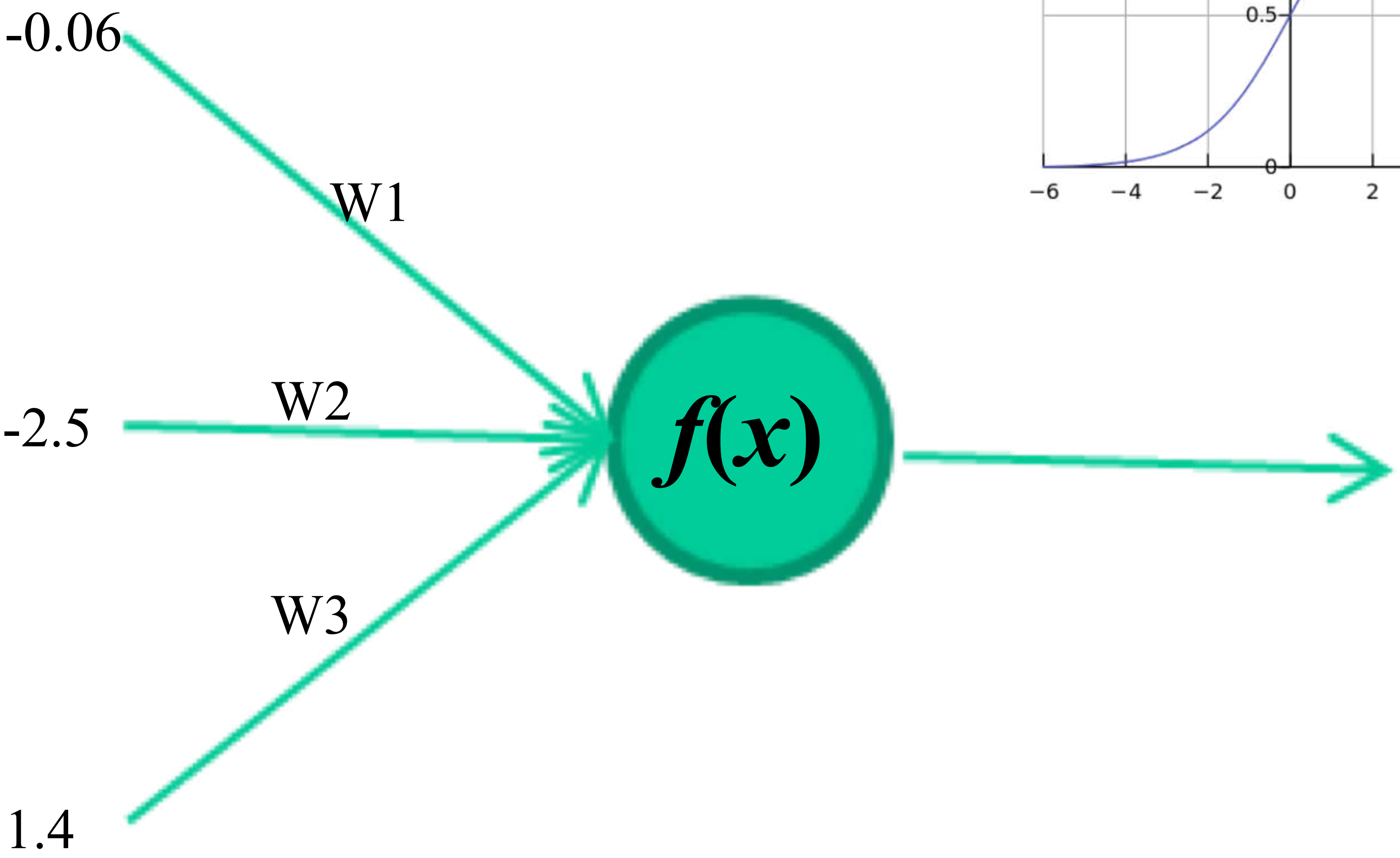
$$XOR\ Gate\ using\ step\ function\ with\ Threshold\ (Y) = 2$$



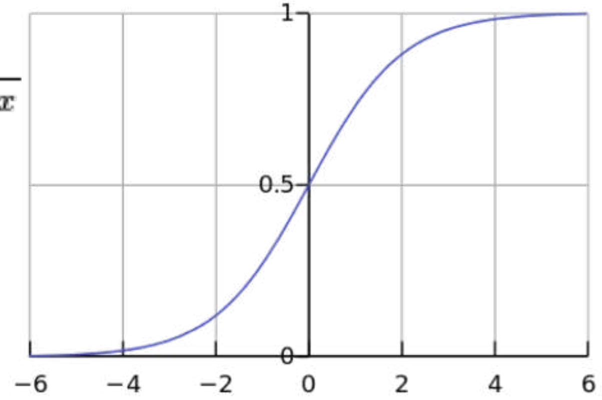
SIGMOID ACTIVATION

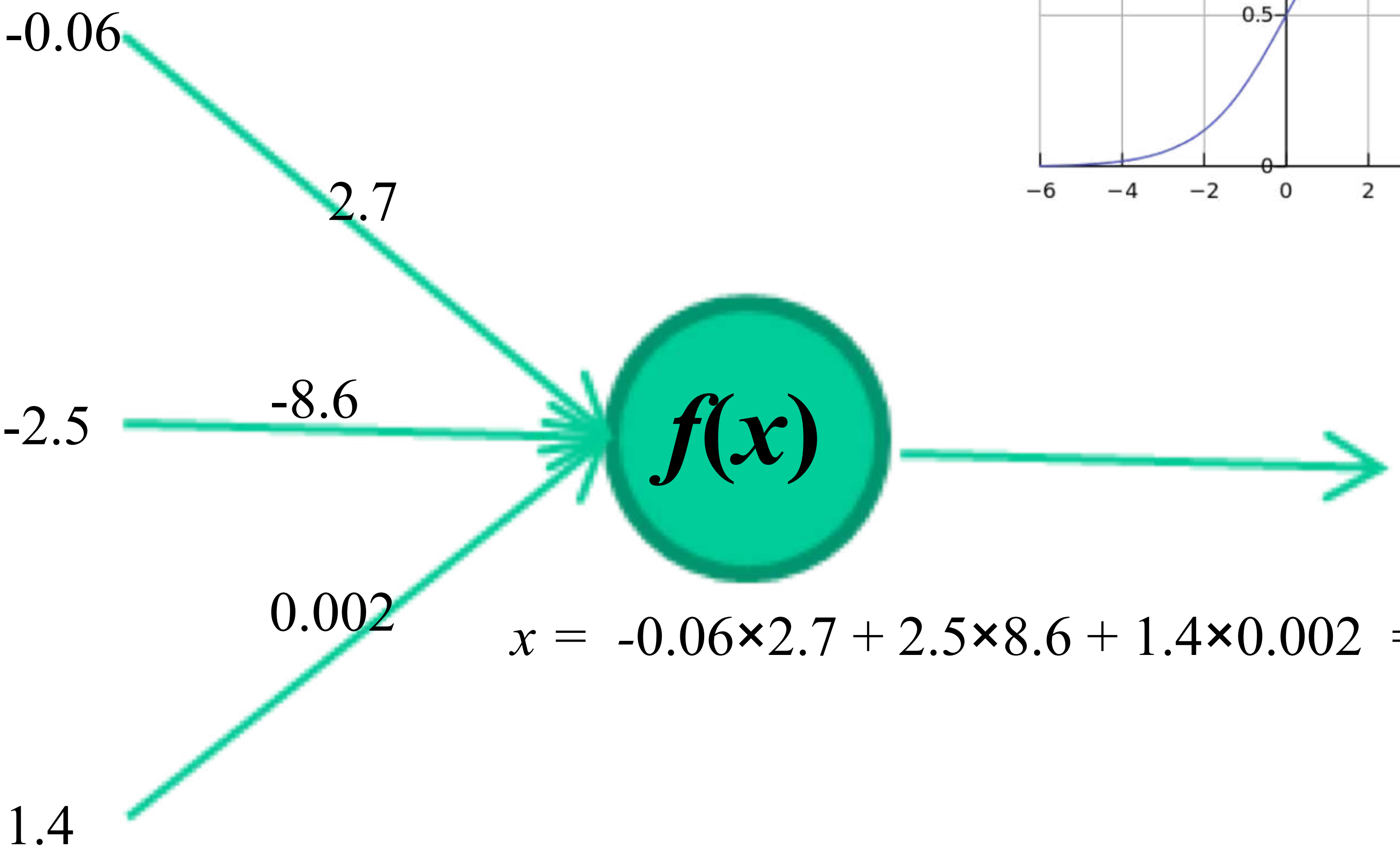
$$f(x) = \frac{1}{1 + e^{-x}}$$



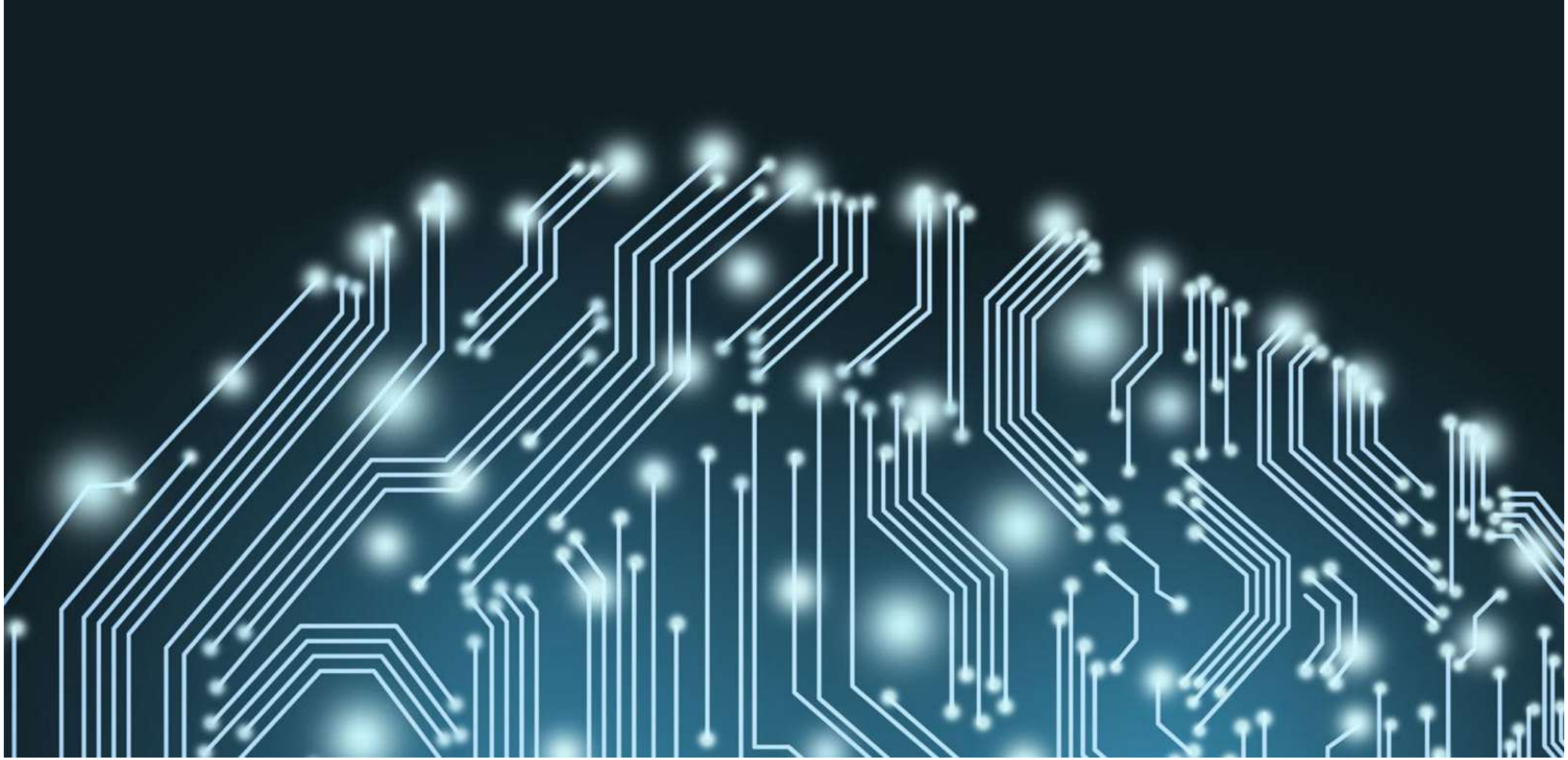


$$f(x) = \frac{1}{1 + \mathrm{e}^{-x}}$$





$x =$ -0.06×2.7 + 2.5×8.6 + 1.4×0.002 = 21.34

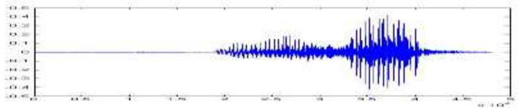


COMBINING ARTIFICIAL NEURONS

DEEP LEARNING OUTLINE

- Introduction of Deep Learning
- “Hello World” for Deep Learning
- Tips for Deep Learning

MACHINE LEARNING
≈ LOOKING FOR A FUNCTION

- Speech Recognition $f($  $)=$ “How are you”
- Image Recognition $f($  $)=$ “Cat”
- Playing Go $f($  $)=$ “5-5” (next move)
- Dialogue System $f($ “Hi” $)=$ “Hello”

  (what the user said) (system response)

THREE STEPS FOR DEEP LEARNING

Step 1: define a set of function

Neural Network

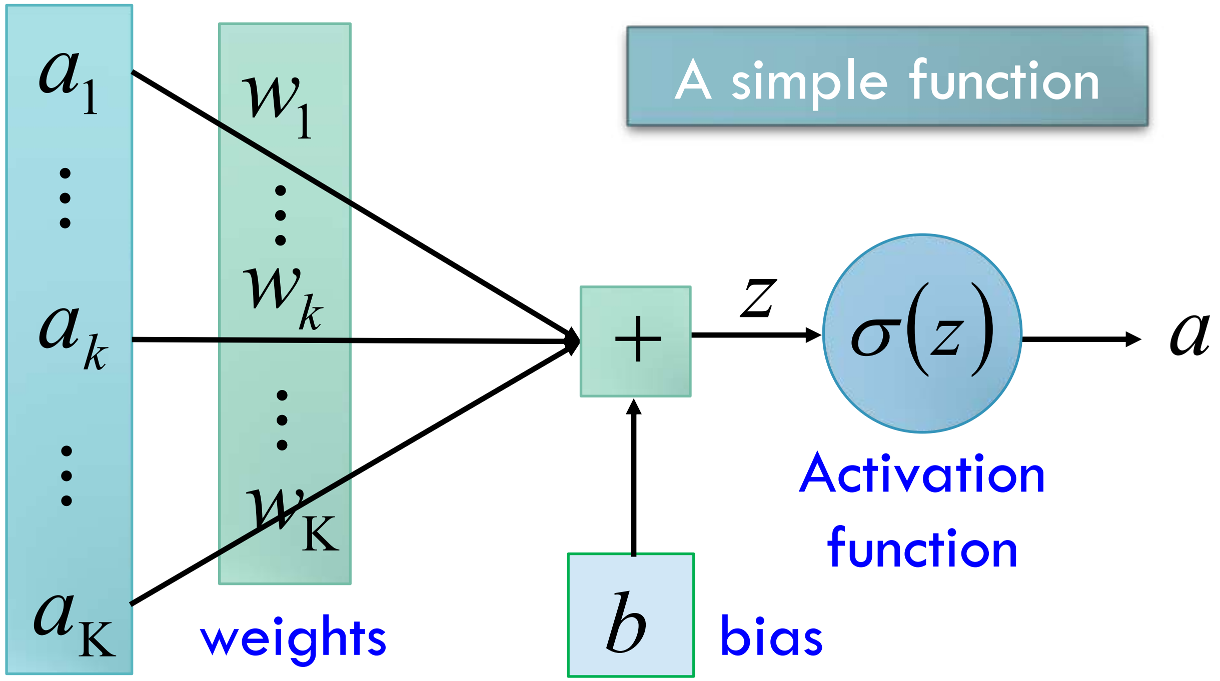
Step 2: goodness of function

Step 3: pick the best function

NEURAL NETWORK

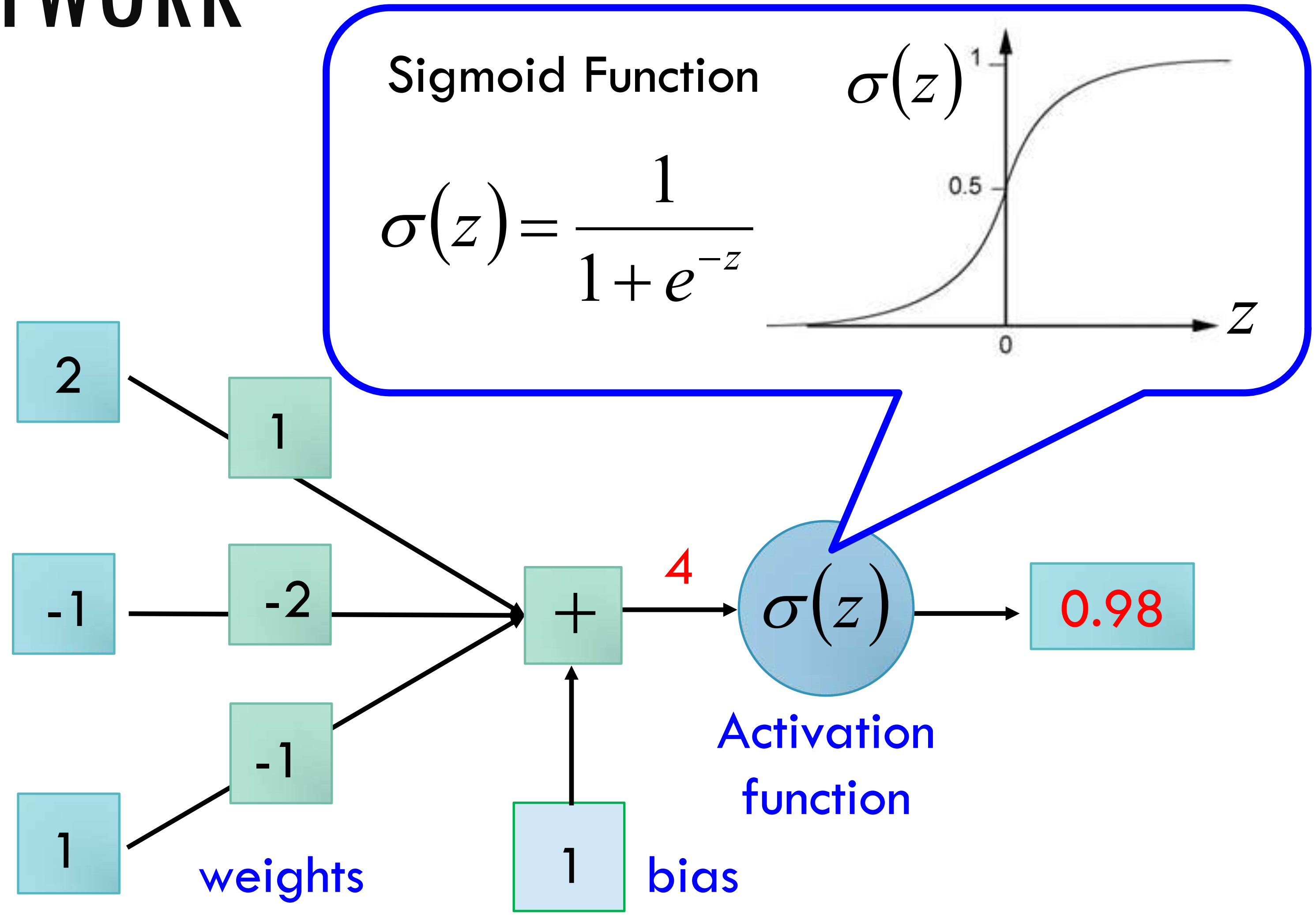
Neuron

$$z = a_1 w_1 + \cdots + a_k w_k + \cdots + a_K w_K + b$$



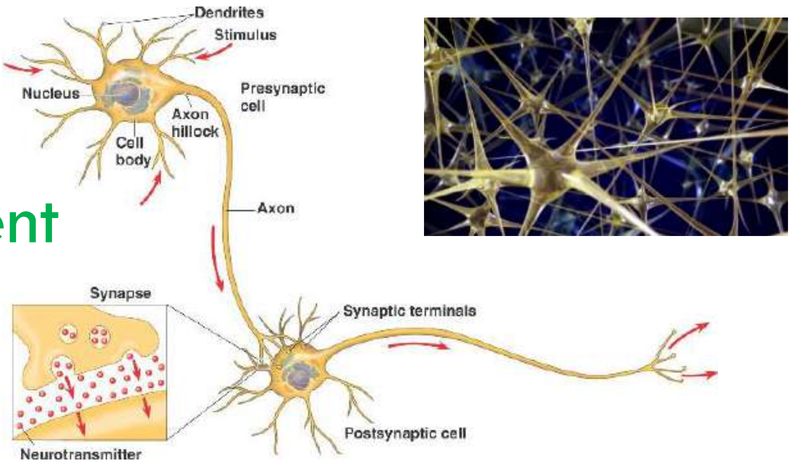
NEURAL NETWORK

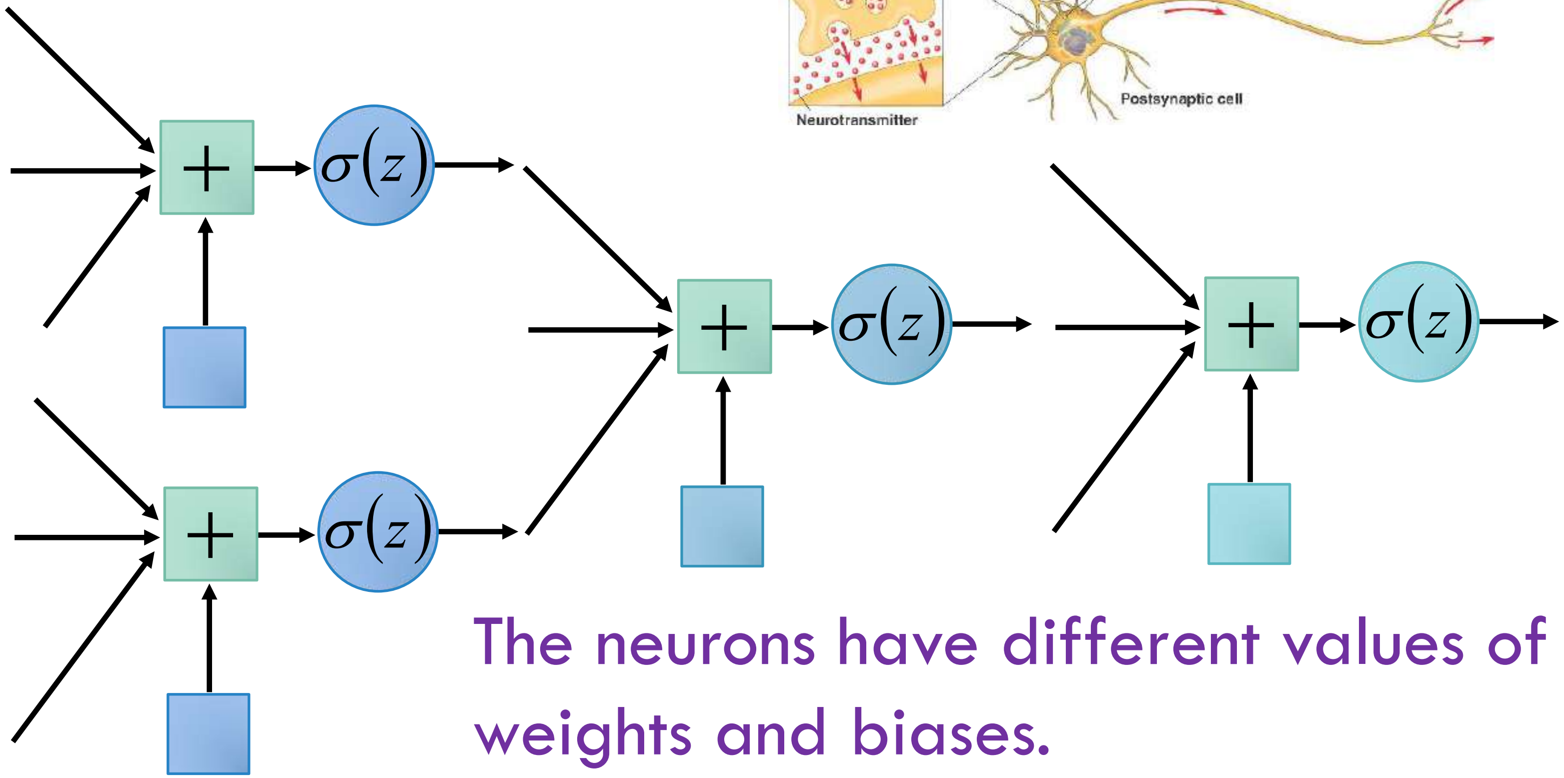
Neuron



NEURAL NETWORK

Different connections lead to different network structures

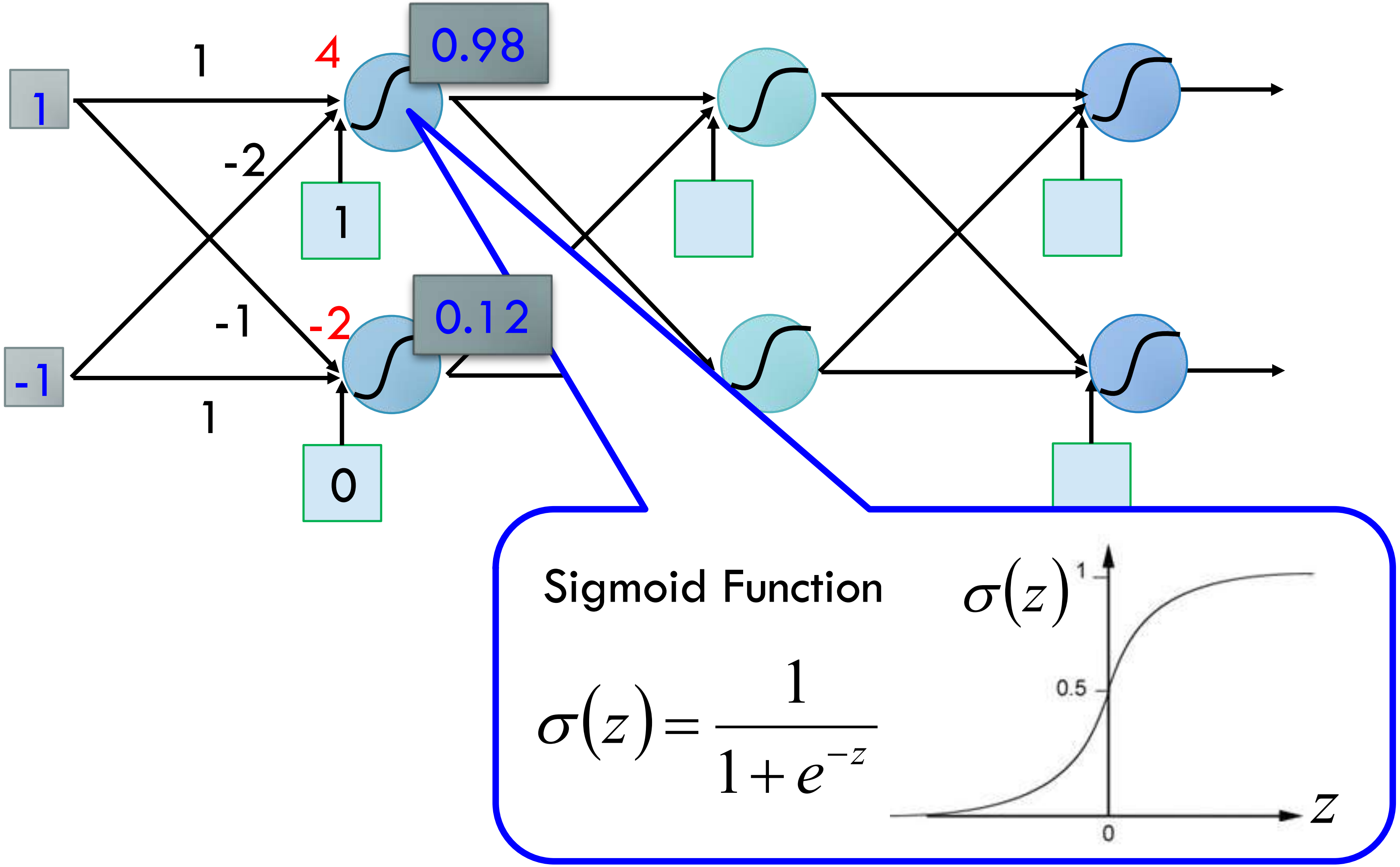




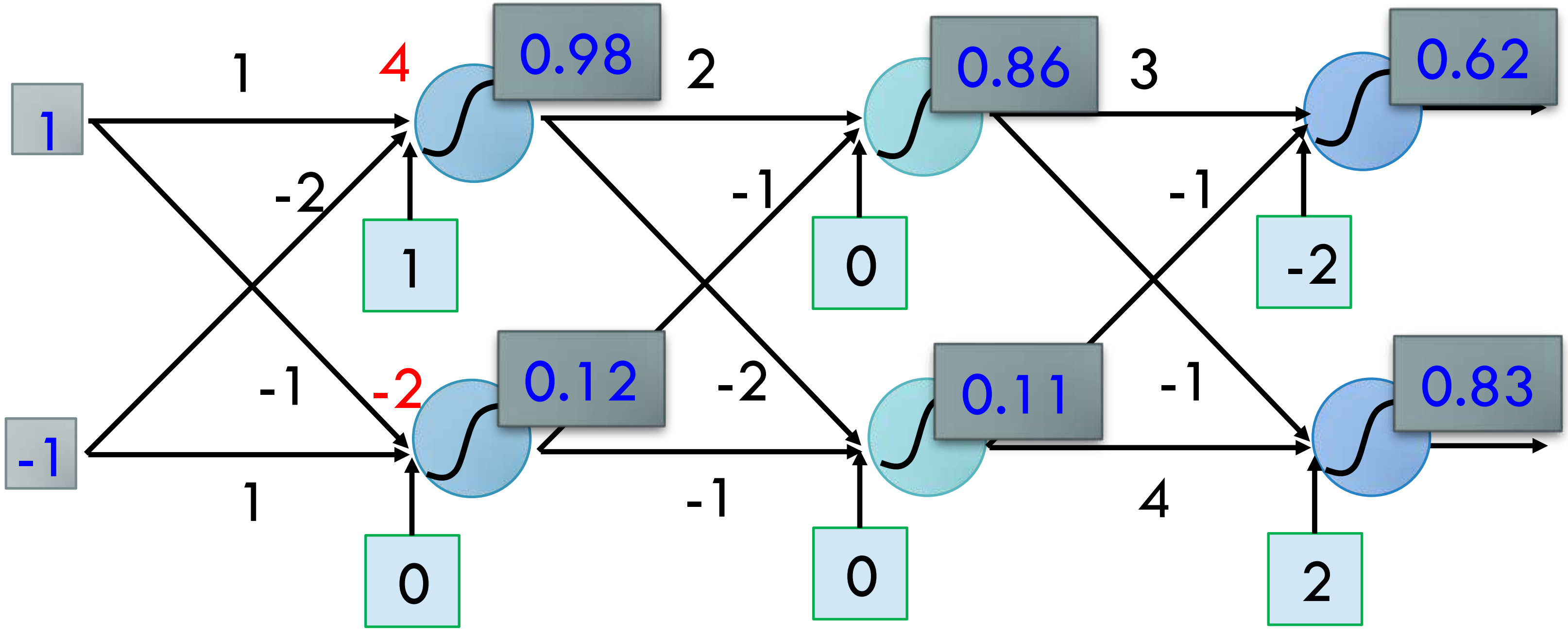
The neurons have different values of weights and biases.

Weights and biases are network parameters θ

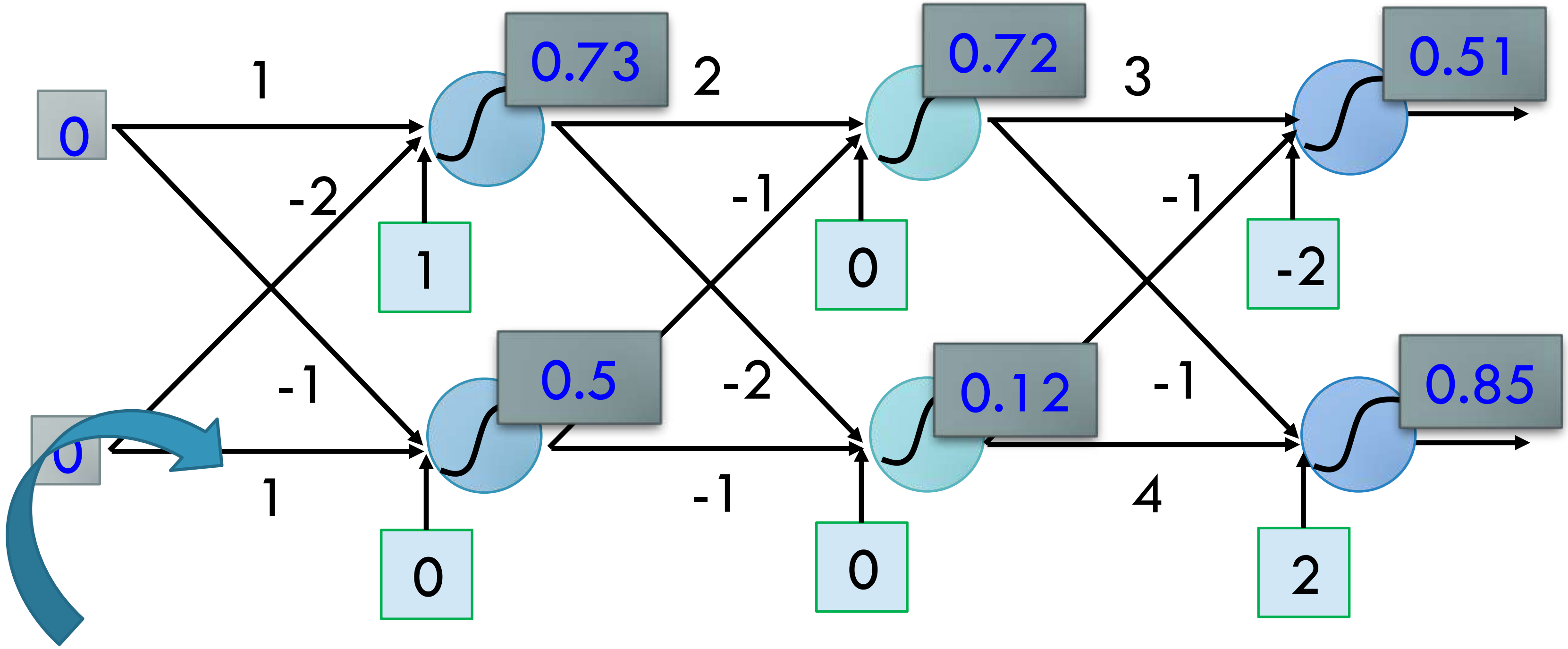
FULLY CONNECT FEEDFORWARD NETWORK



FULLY CONNECT FEEDFORWARD NETWORK



FULLY CONNECT FEEDFORWARD NETWORK



This is a function.

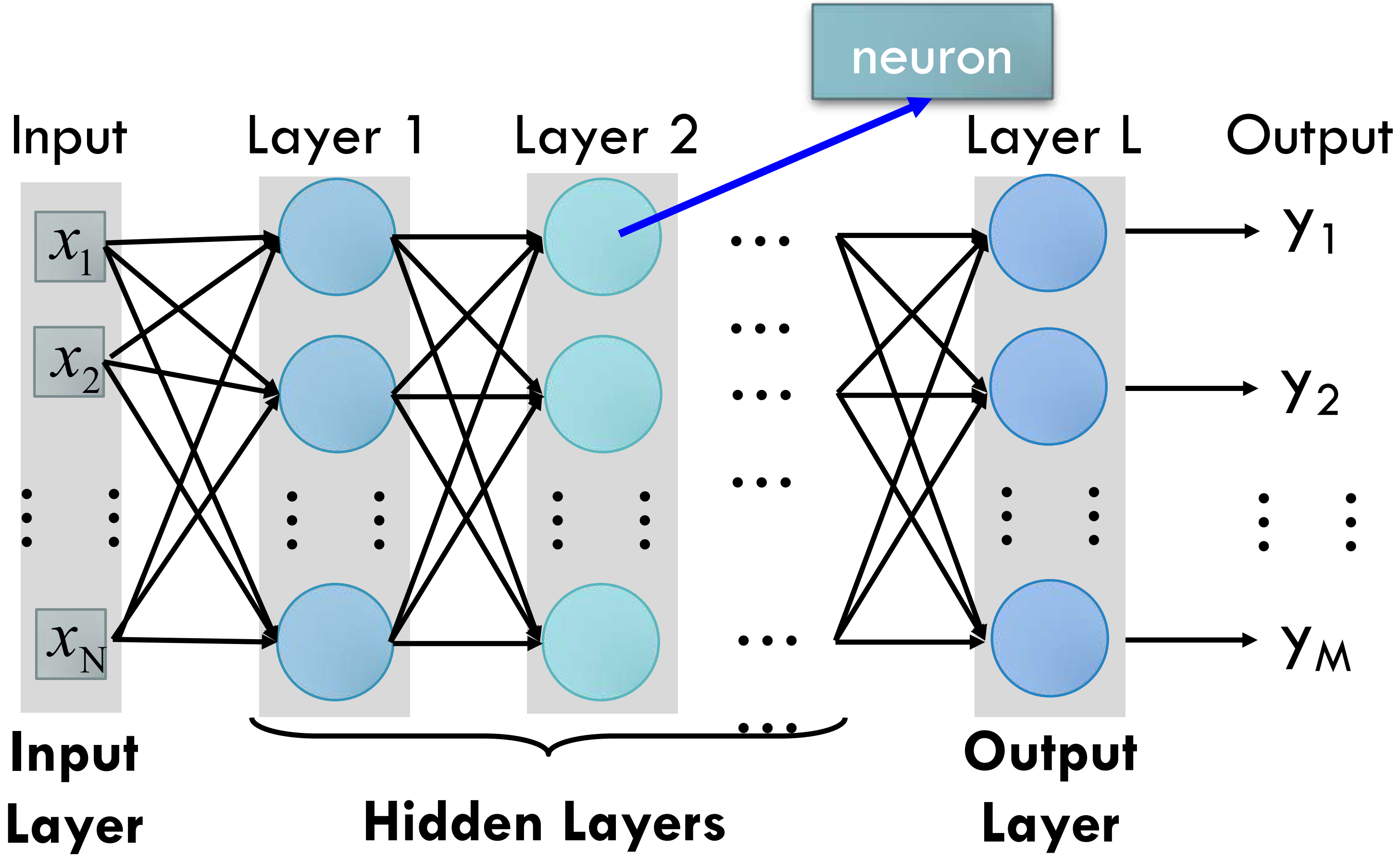
Input vector, output vector

$$f\left(\begin{bmatrix}1\\-1\end{bmatrix}\right)=\begin{bmatrix}0.62\\0.83\end{bmatrix}\quad f\left(\begin{bmatrix}0\\0\end{bmatrix}\right)=\begin{bmatrix}0.51\\0.85\end{bmatrix}$$

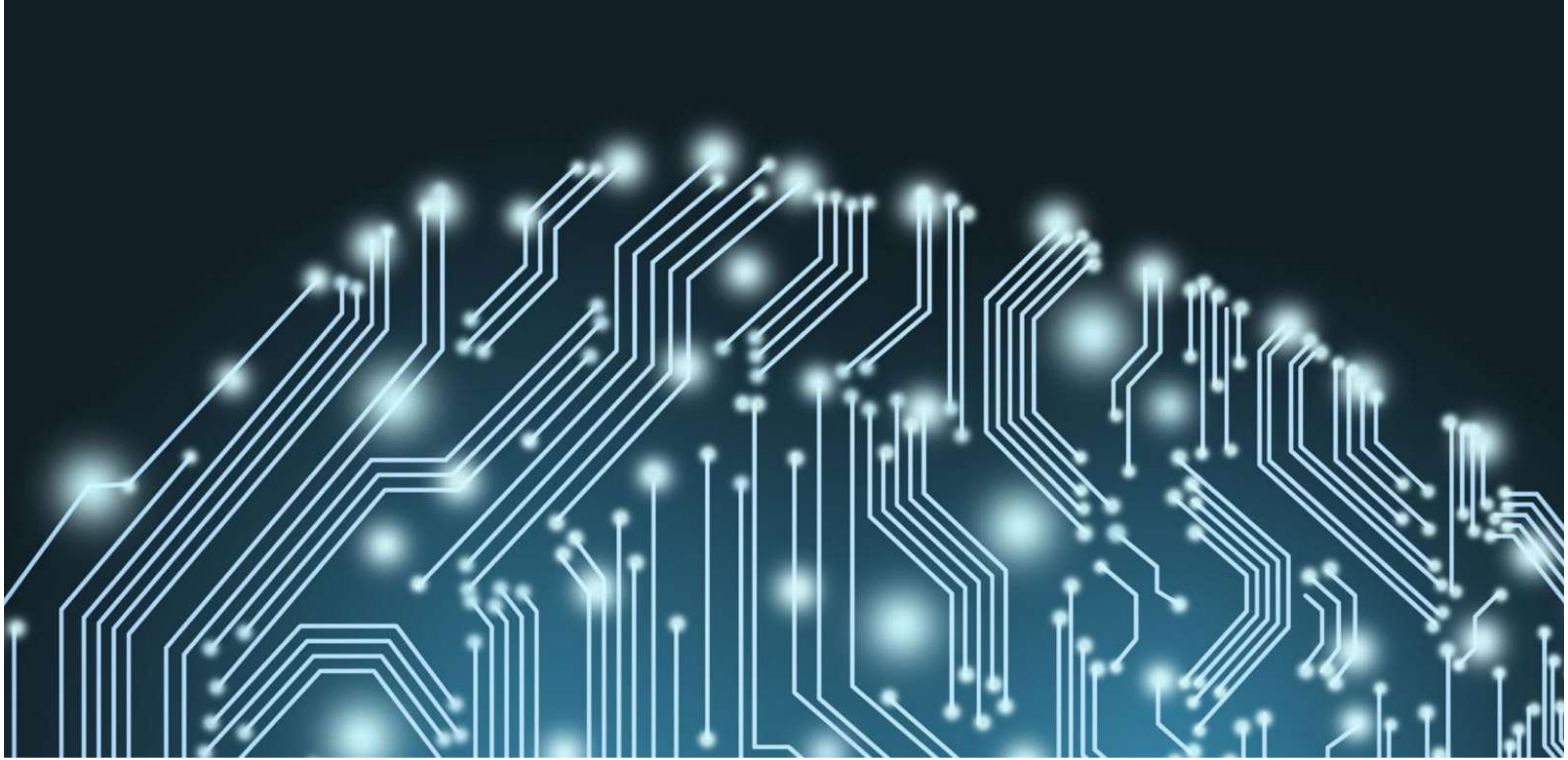
Given parameters θ, define a function

Given network structure, define ***a function set***

FULLY CONNECT FEEDFORWARD NETWORK



Deep means many hidden layers



EXAMPLE: DIGIT CLASSIFICATION

FEATURE DETECTORS

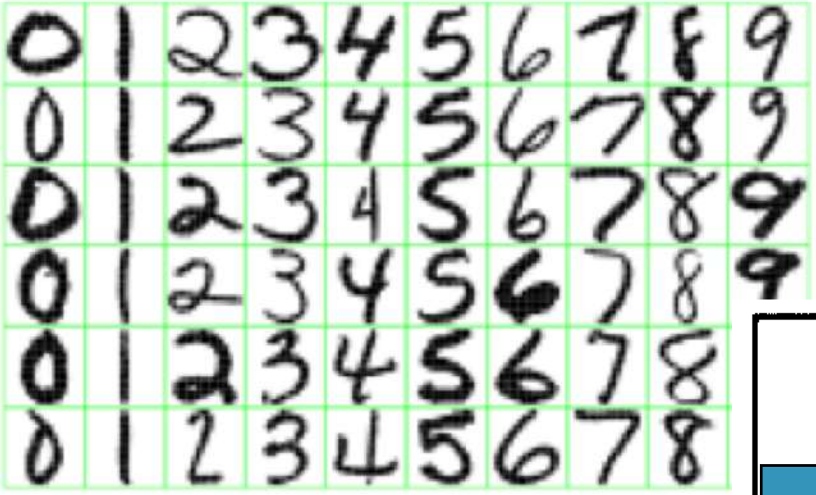
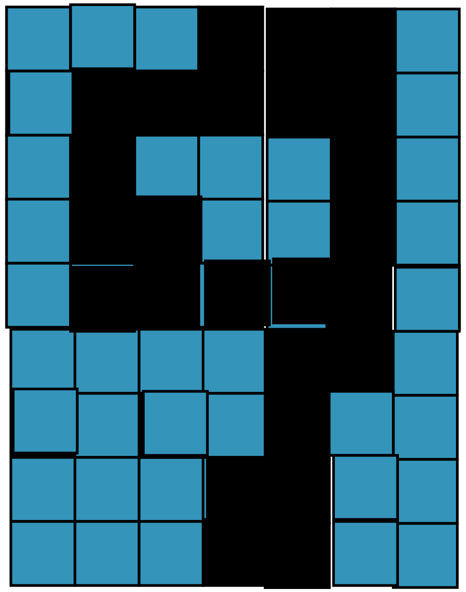
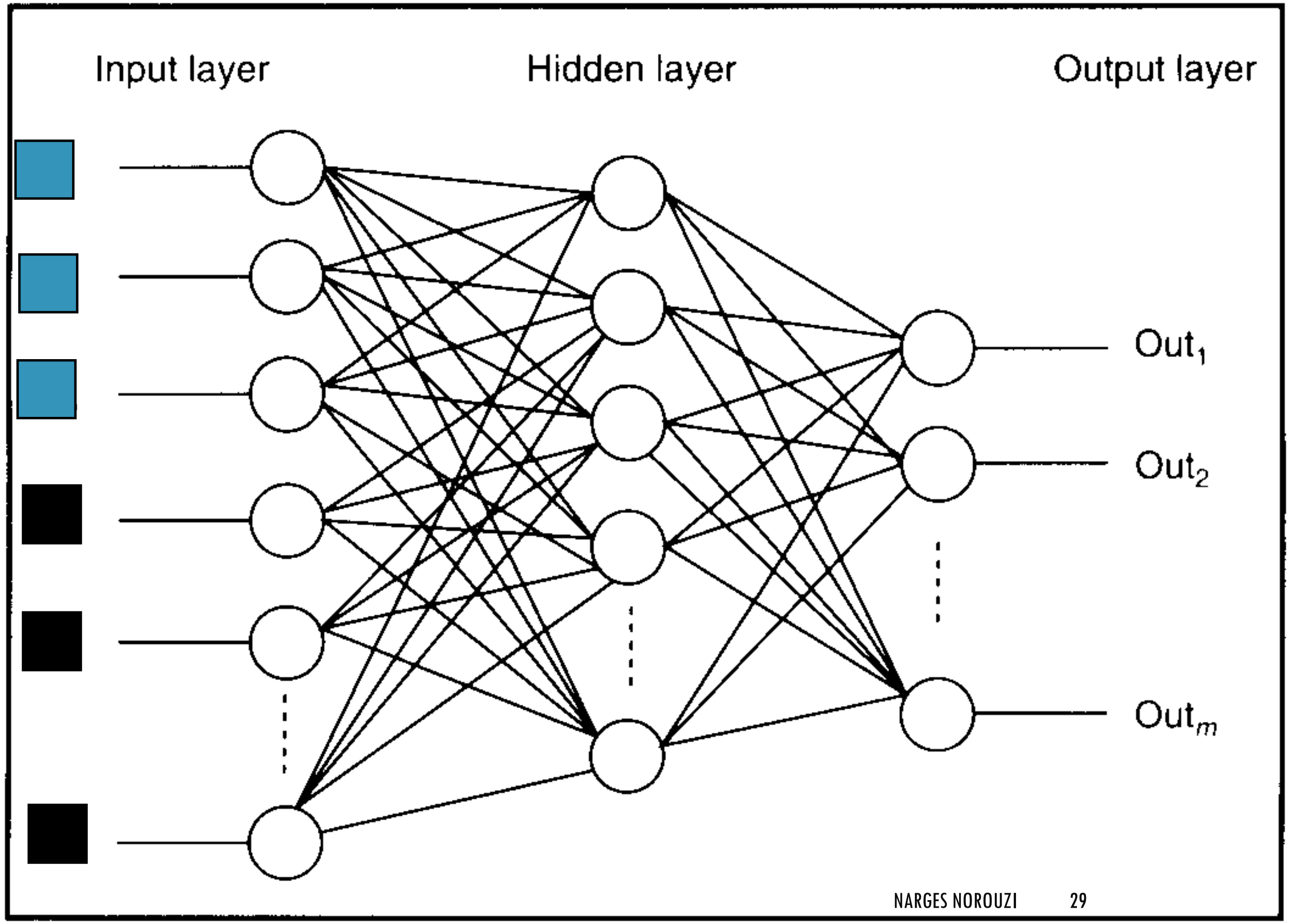


Figure 1.2: *Examples of handwritten digits from postal envelopes.*





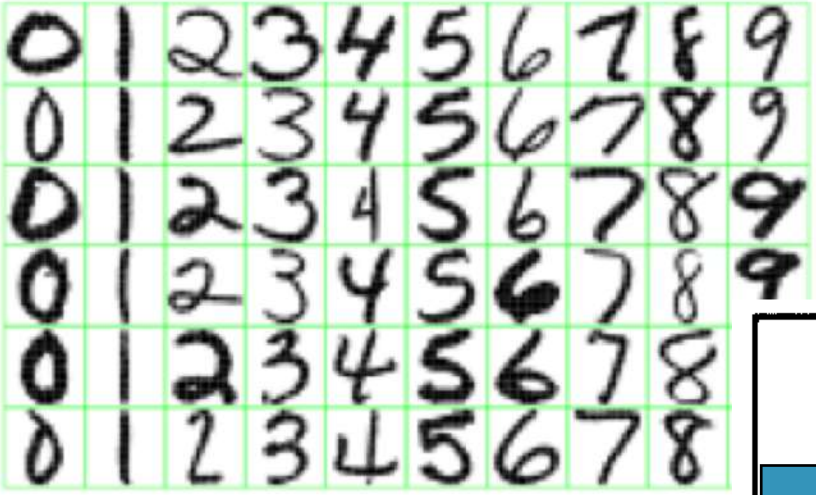
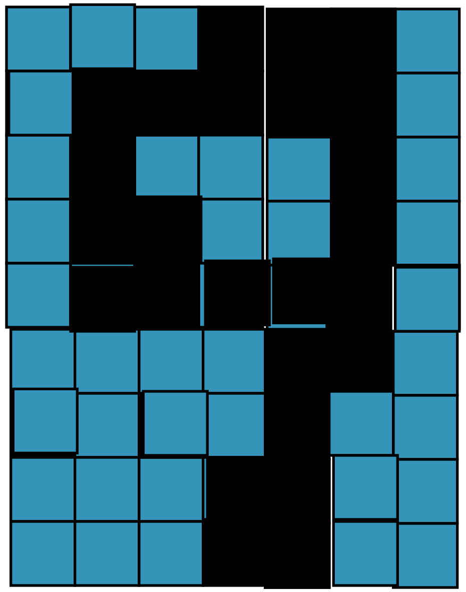
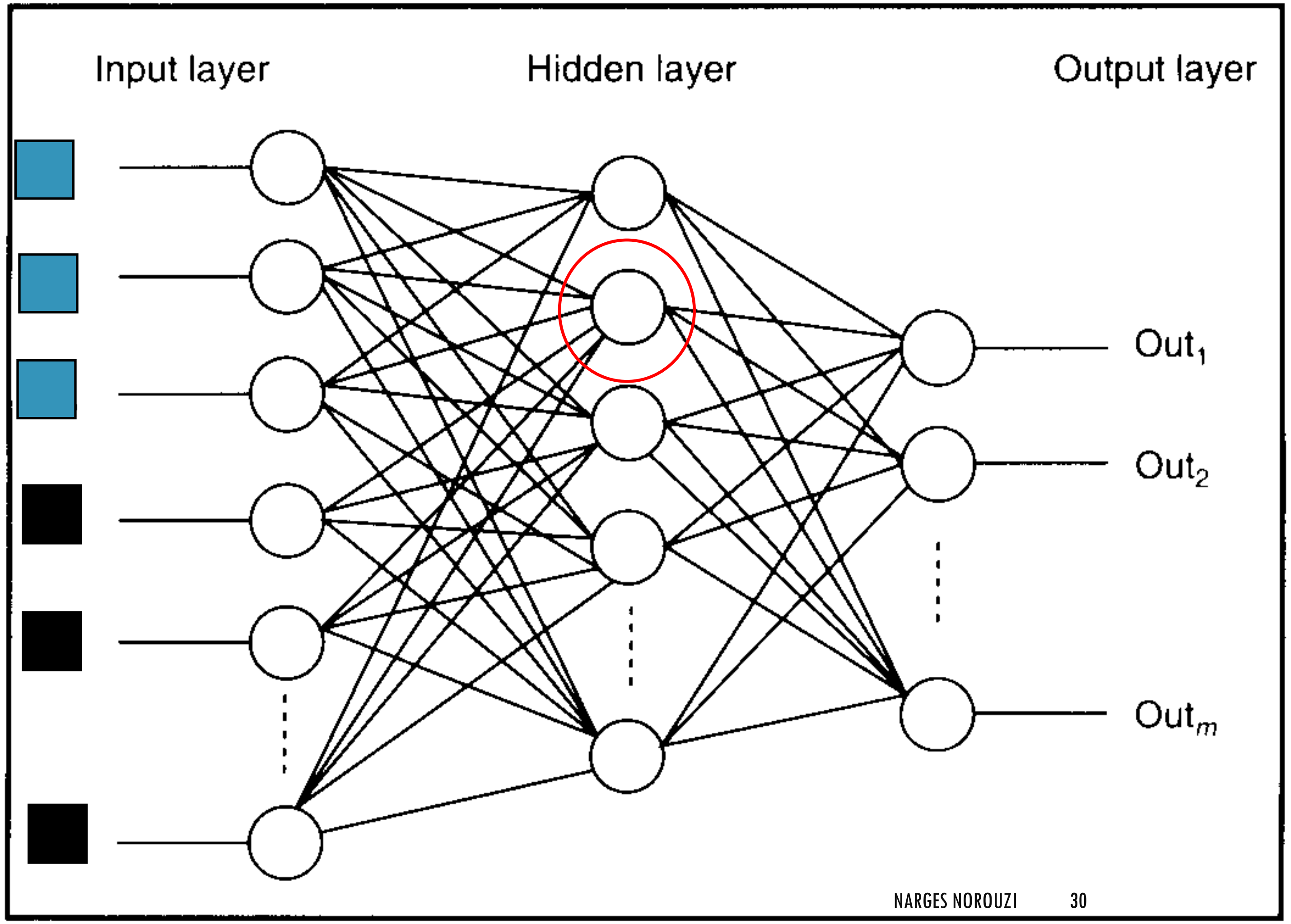


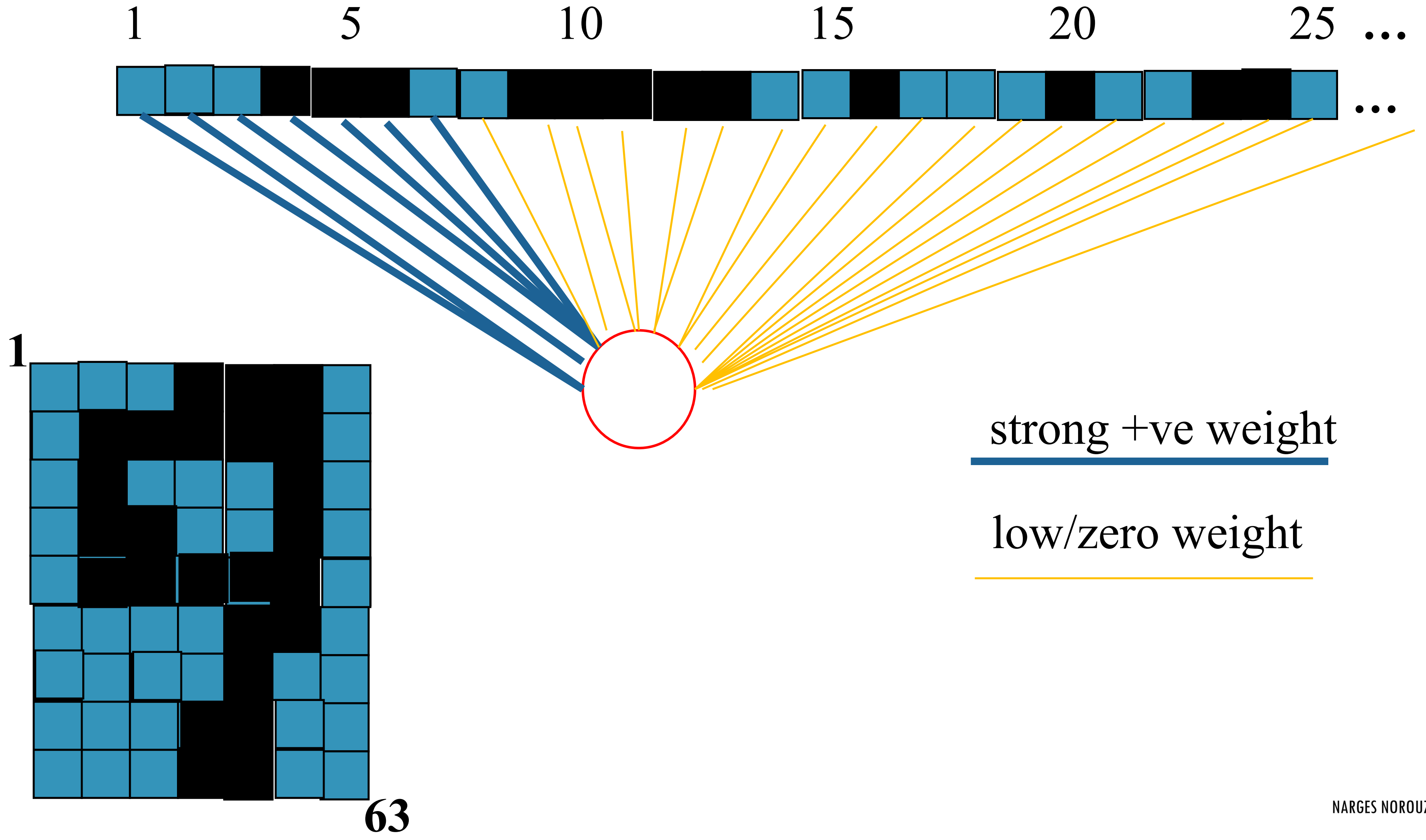
Figure 1.2: *Examples of handwritten digits from postal envelopes.*



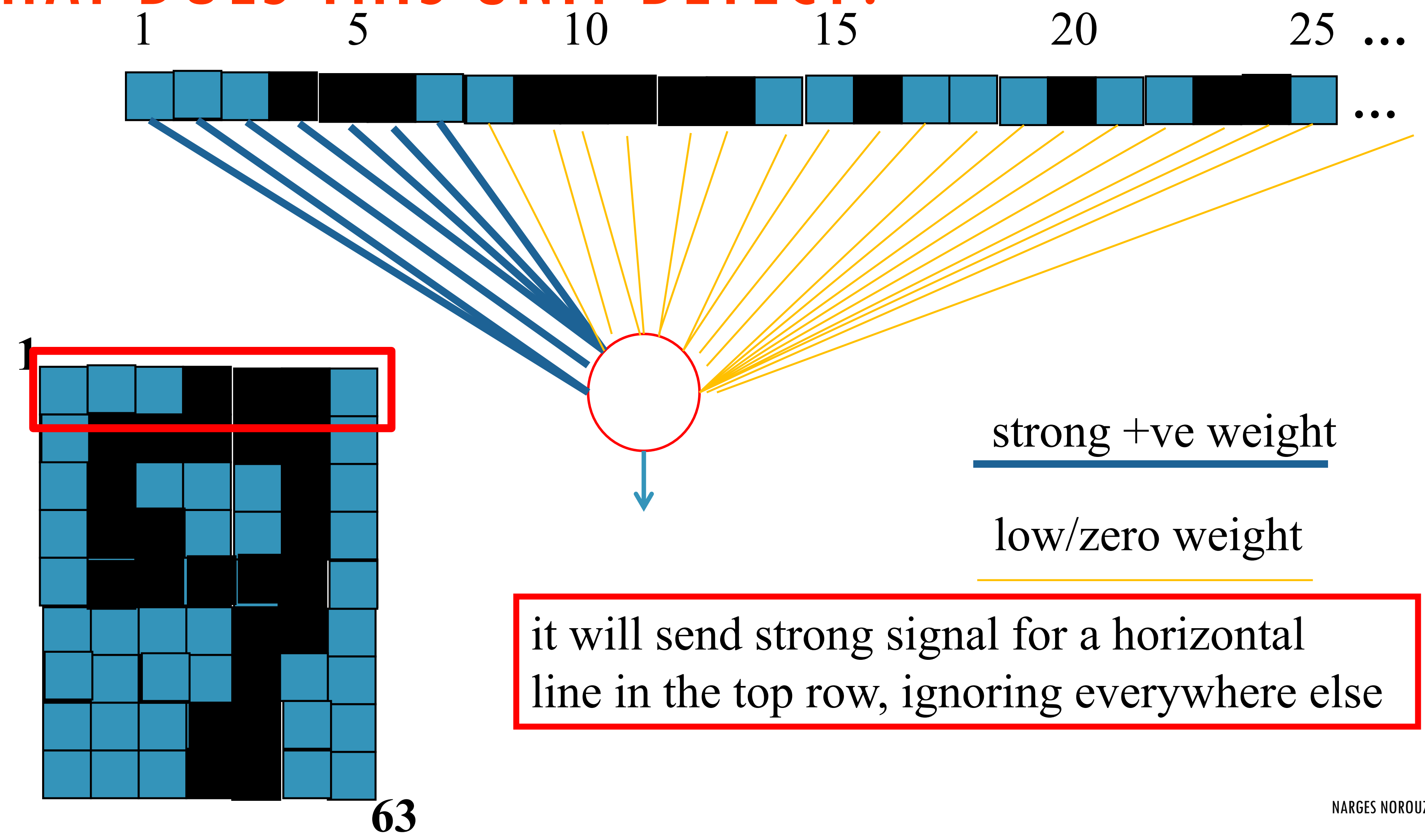
WHAT IS THIS UNIT DOING?



SELF-ORGANIZED FEATURE DETECTORS

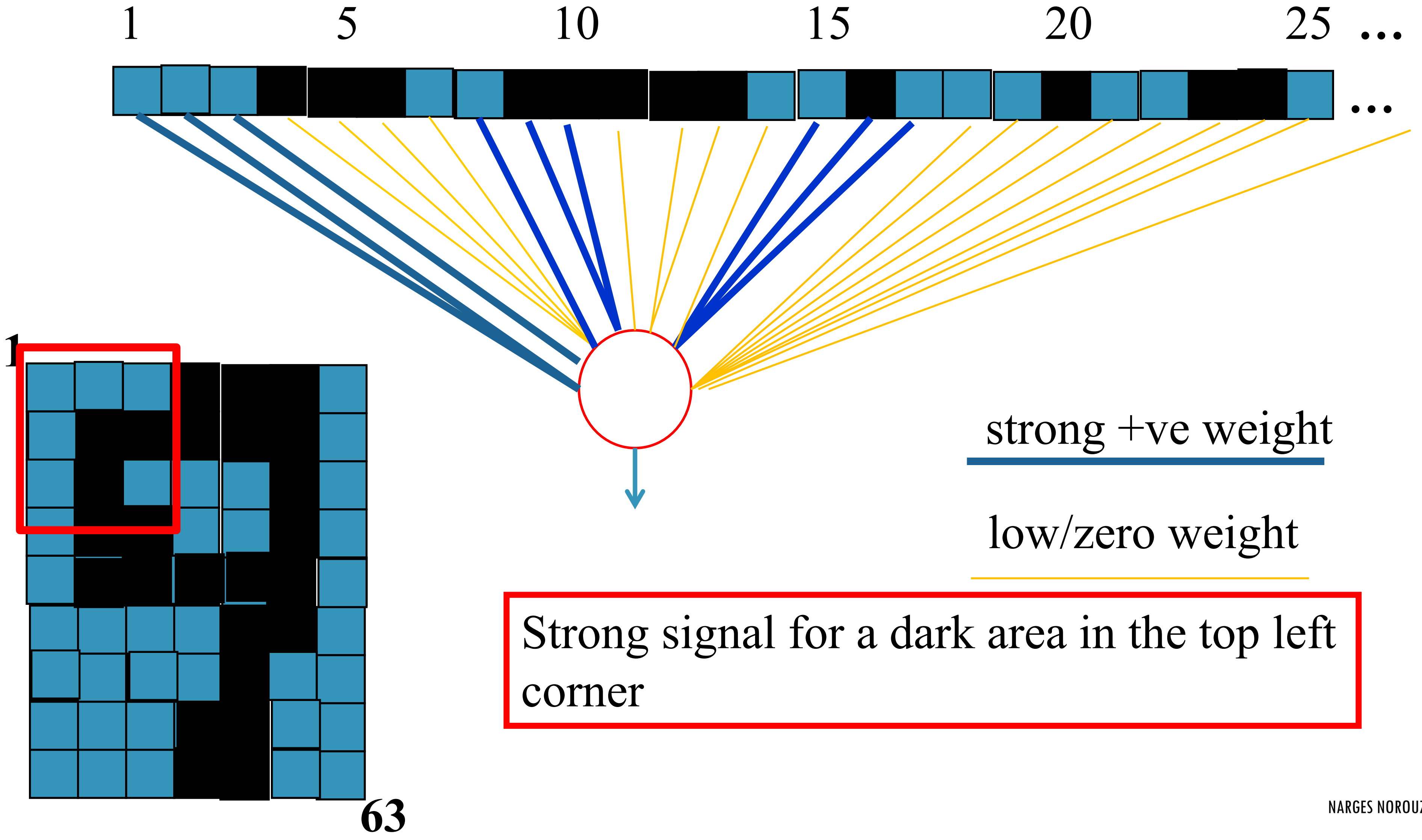


WHAT DOES THIS UNIT DETECT?



it will send strong signal for a horizontal line in the top row, ignoring everywhere else

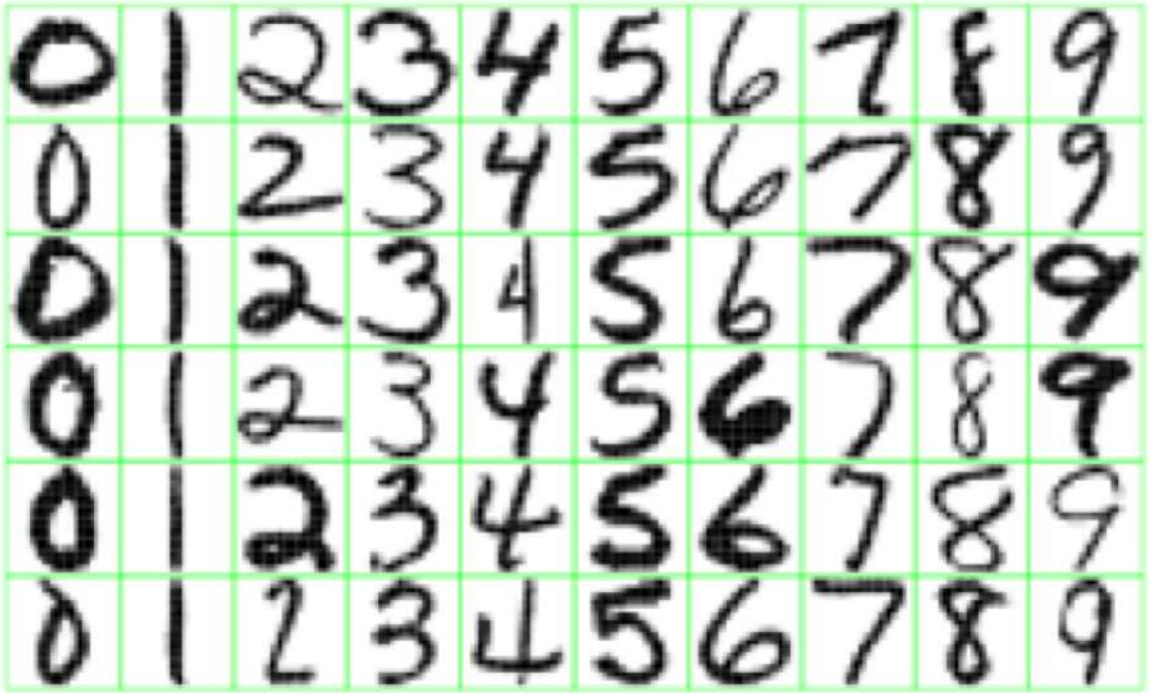
WHAT DOES THIS UNIT DETECT?



strong +ve weight

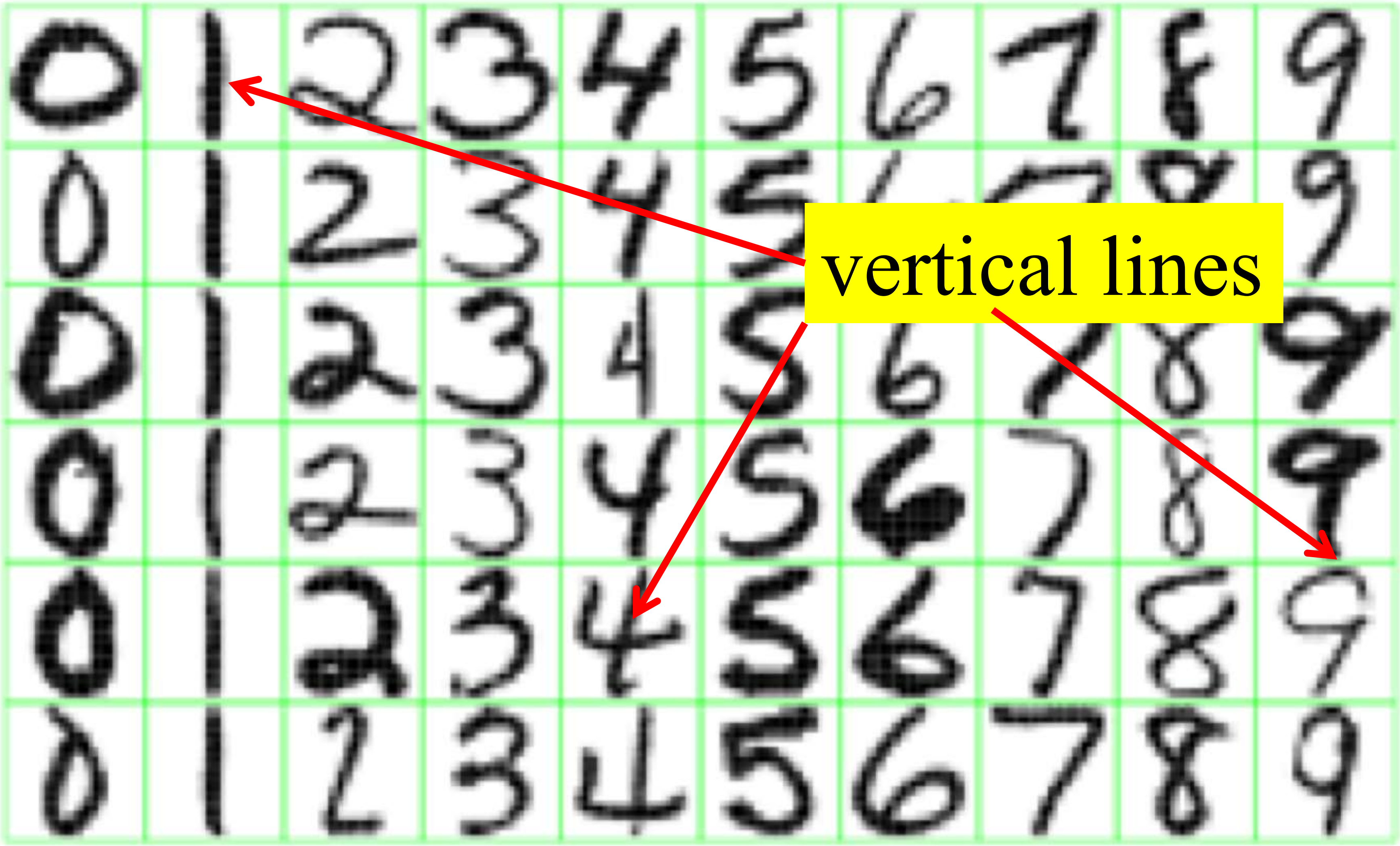
low/zero weight

Strong signal for a dark area in the top left corner

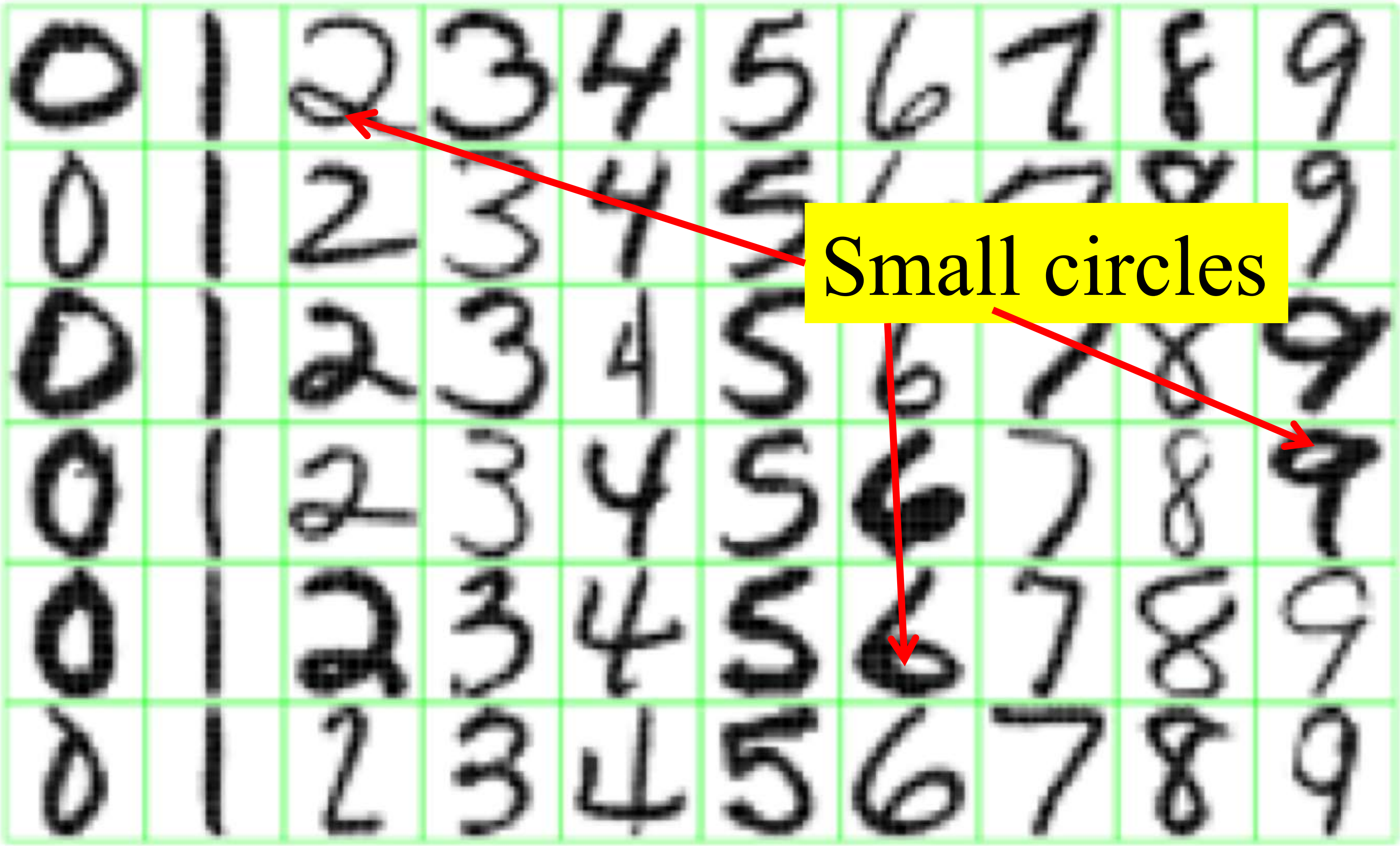


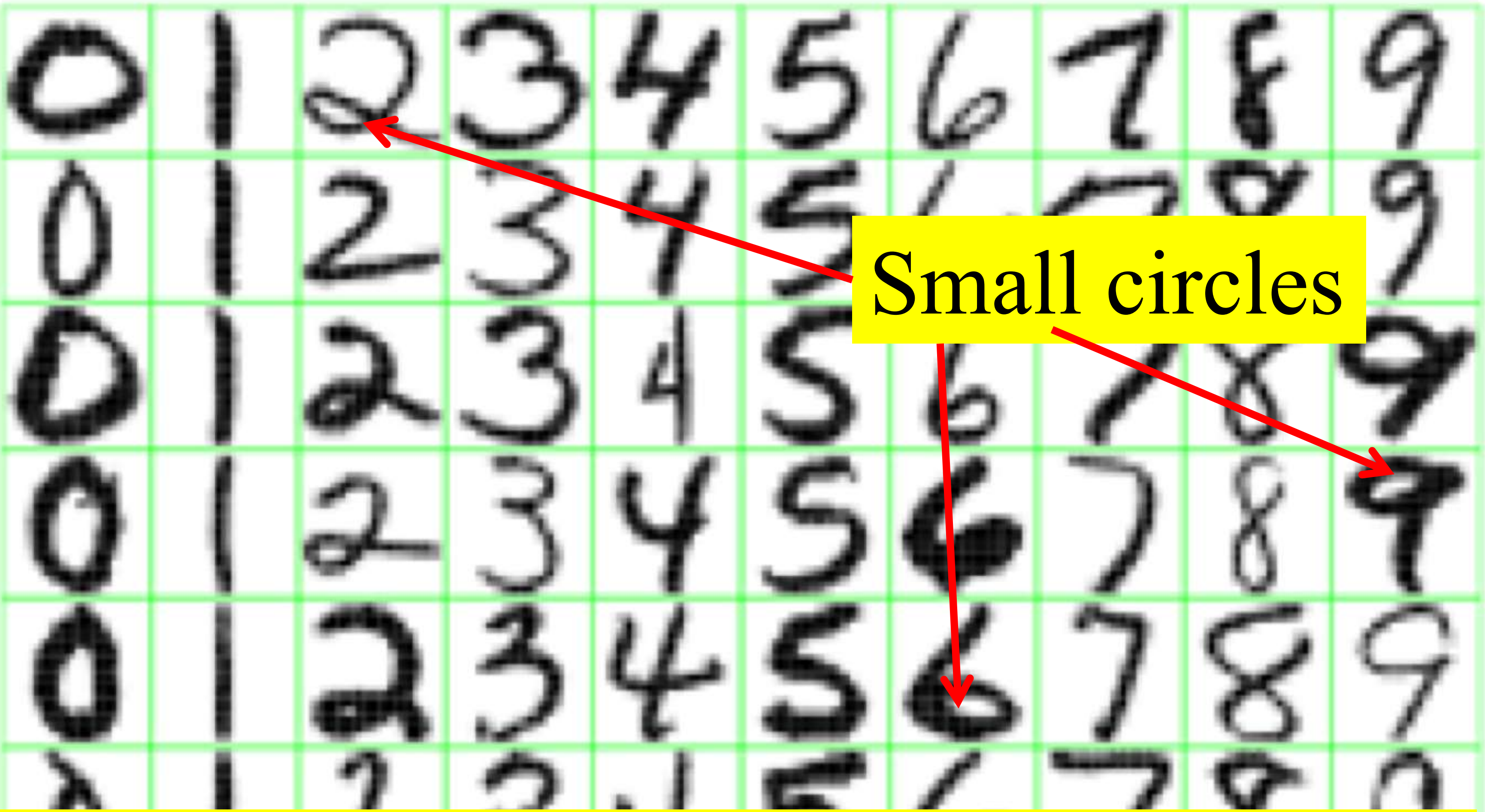
What features might you expect a good NN to learn, when trained with data like this?

1



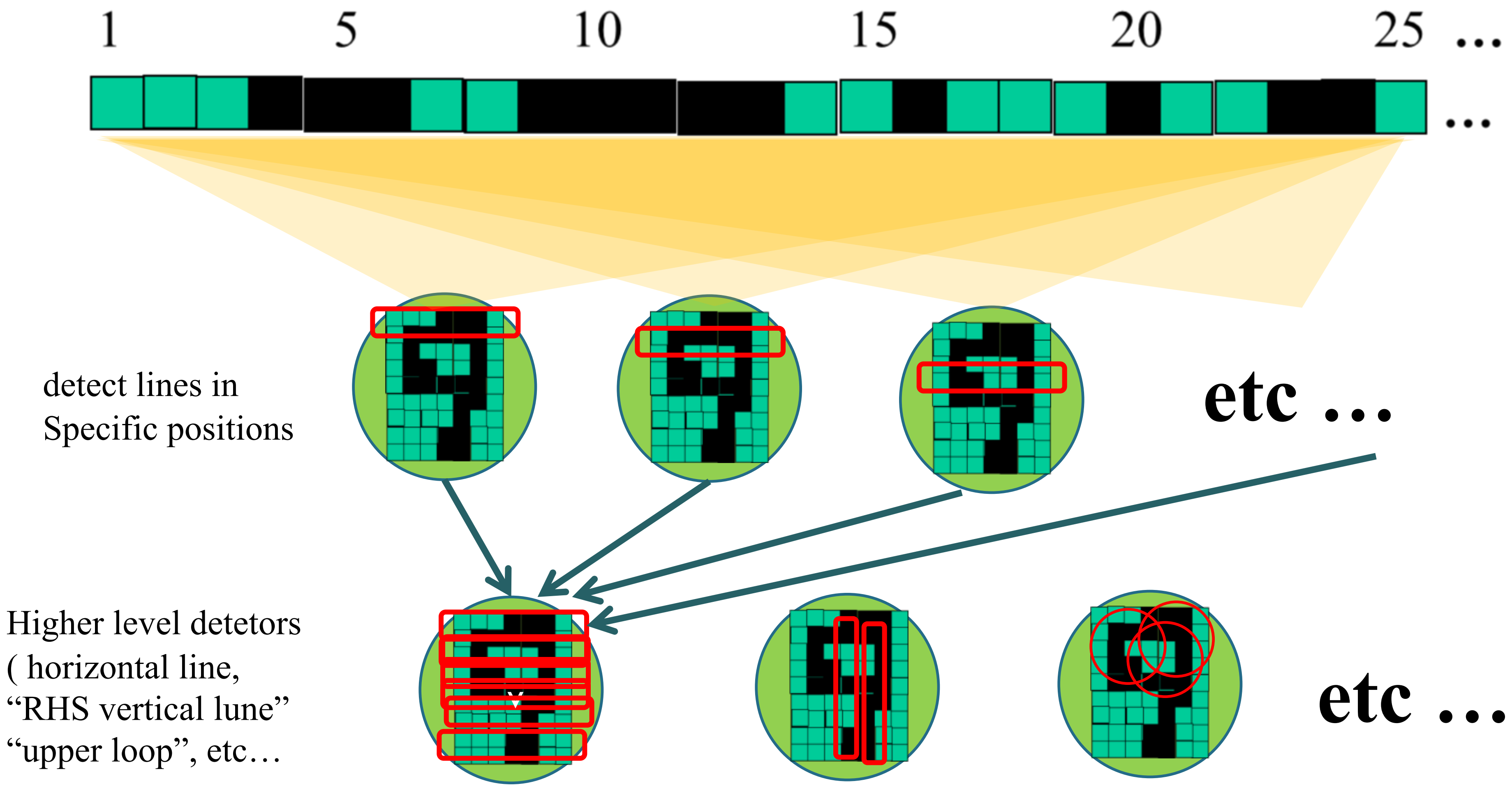

Horizontal lines


Small circles

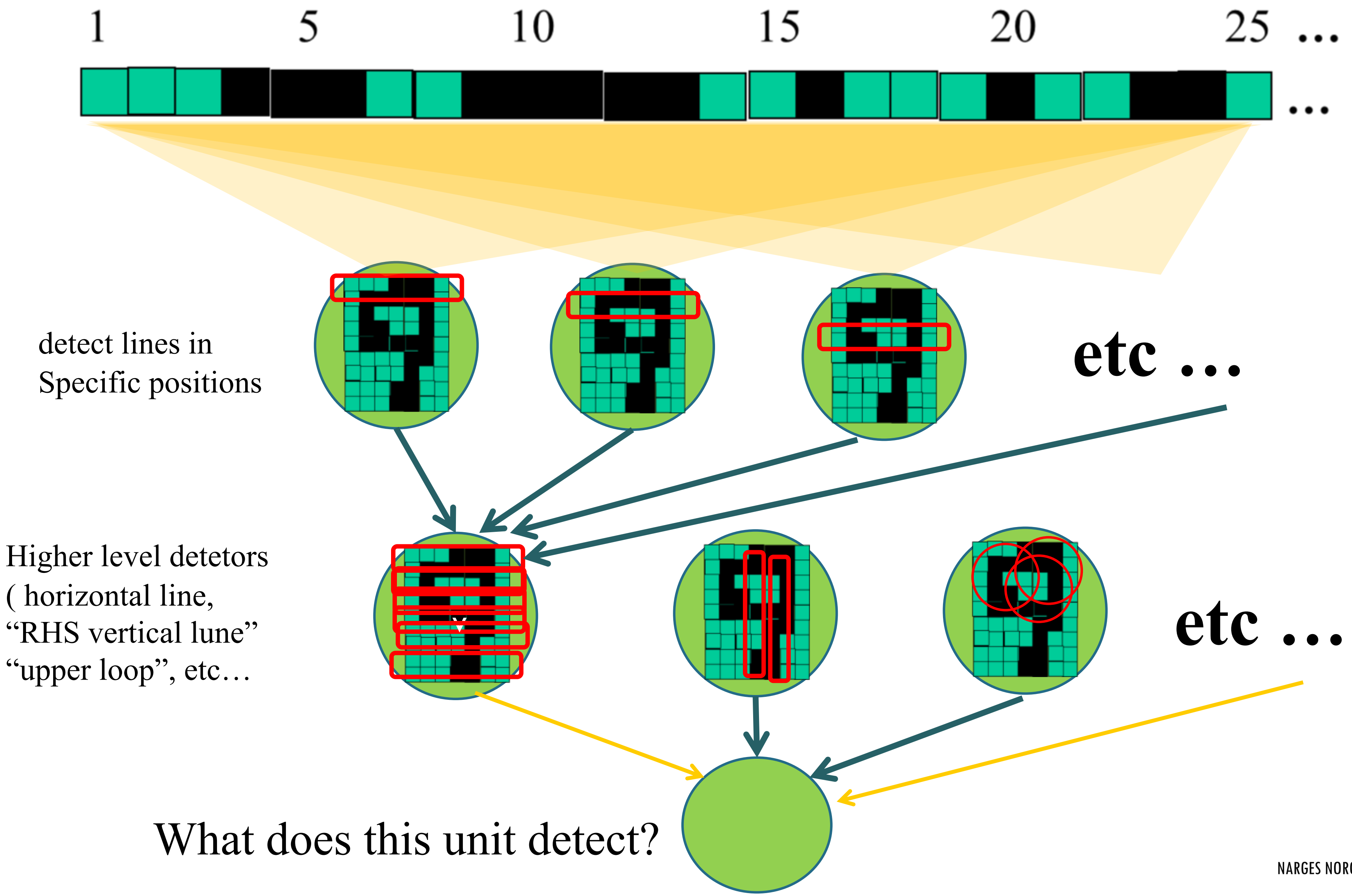


But what about position invariance ???
our example unit detectors were tied to specific parts of the image

SUCCESSIVE LAYERS CAN LEARN HIGHER-LEVEL FEATURES …

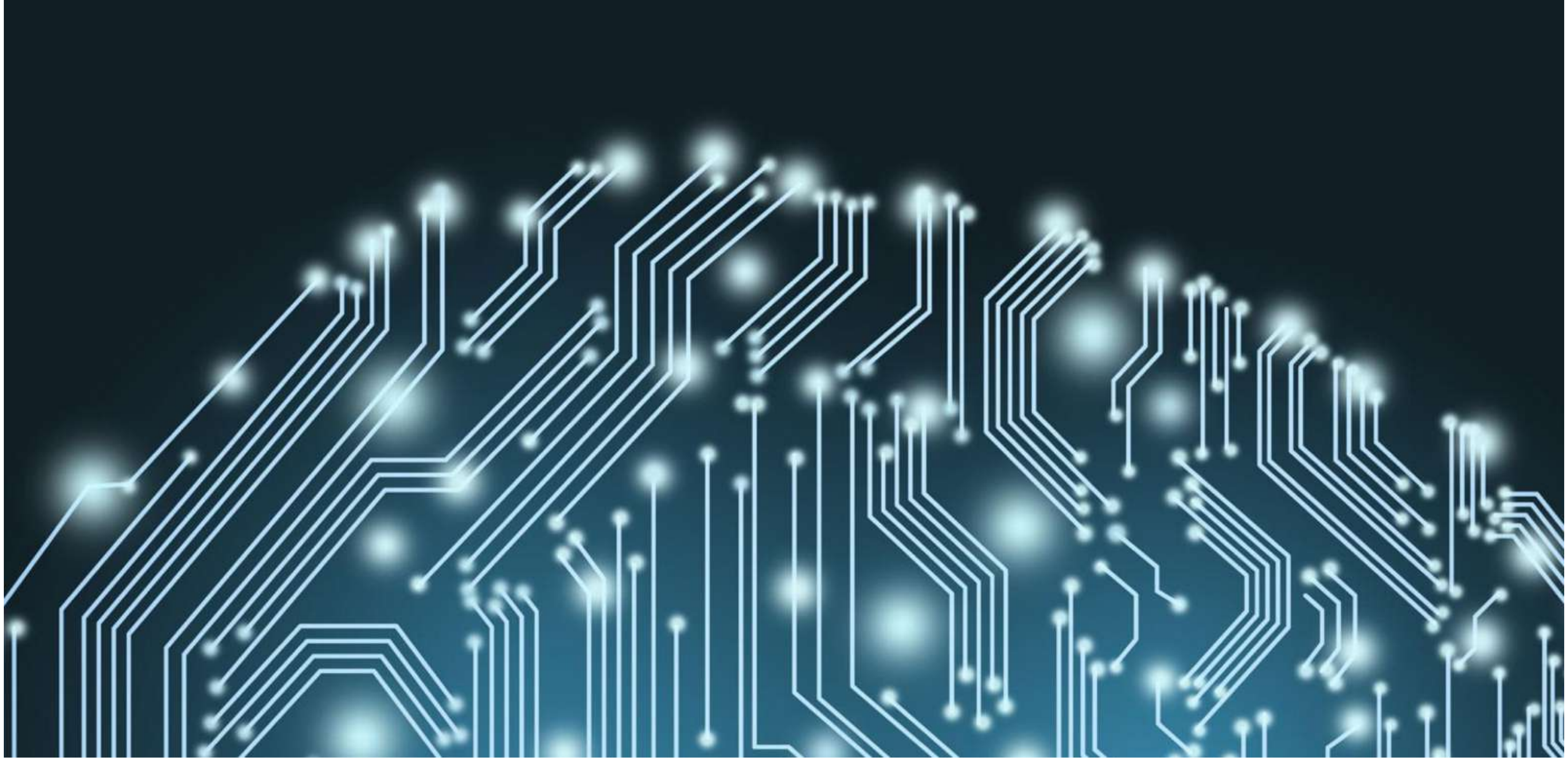


SUCCESSIVE LAYERS CAN LEARN HIGHER-LEVEL FEATURES …



CLASS EXERCISE

bit.ly/ce-10



MORE ABOUT NN LATER…