LECTURE 10

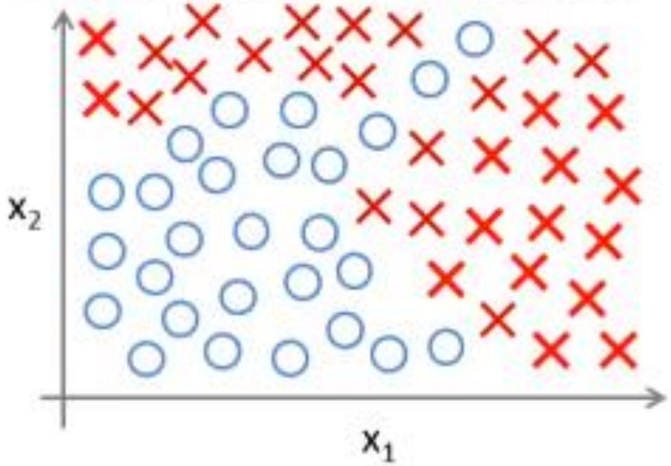
AGENDA FOR TODAY

- Neural Networks and Back Propagation algorithm
- Building your own Neural Network

NEURAL NETWORK

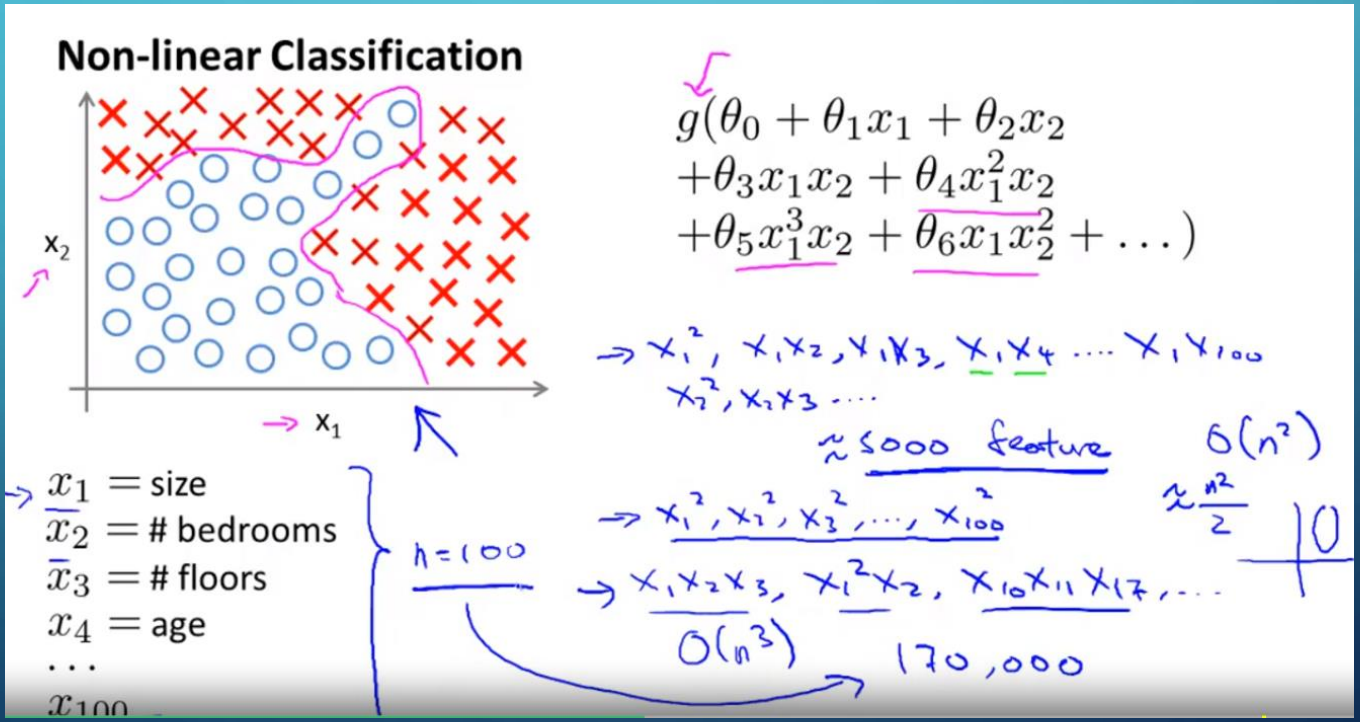
WHY NEURAL NETWORK?

Non-linear Classification

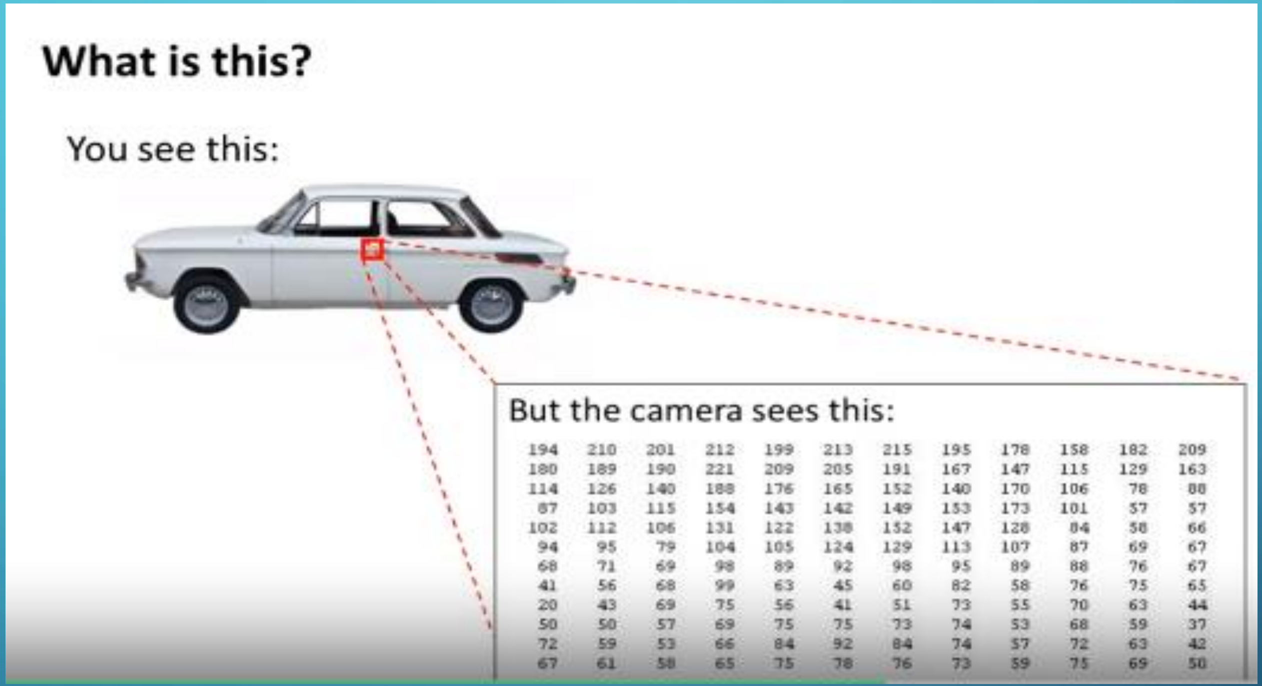


$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2$$
$$+\theta_3 x_1 x_2 + \theta_4 x_1^2 x_2$$
$$+\theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \ldots)$$

- Usually we have lot more features
- If we consider the 2nd degree hypothesis then number of terms will be of the order O(n^2). Similarly for kth degree O(n^k)
- Computationally very expensive and not feasible approach



EXAMPLE OF AN IMAGE



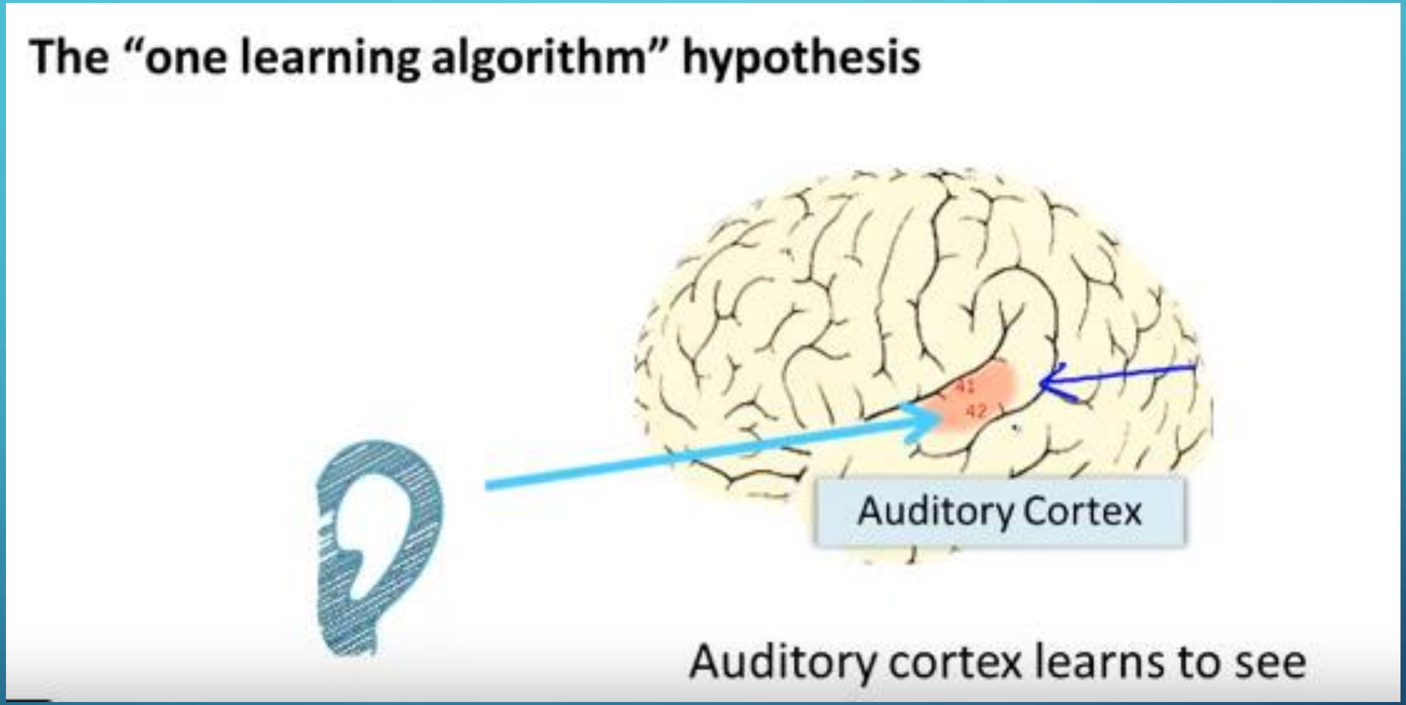
50x50 pixel image will have 2500 features

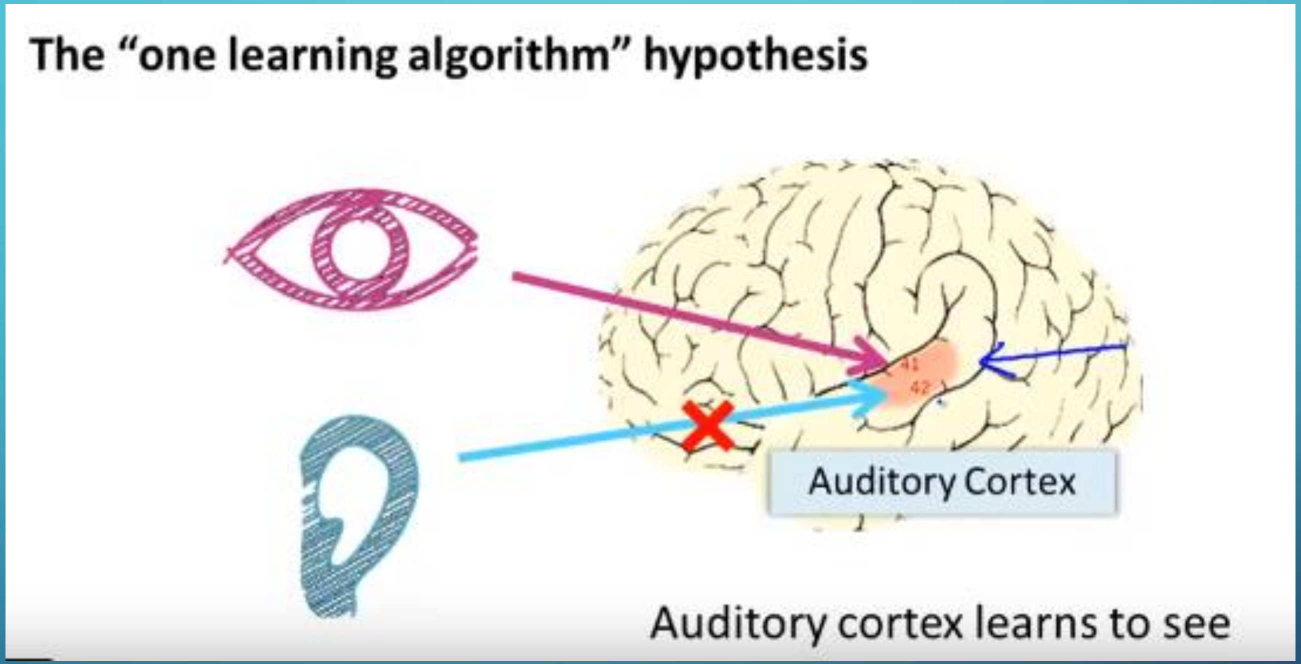
HISTORY

Neural Networks

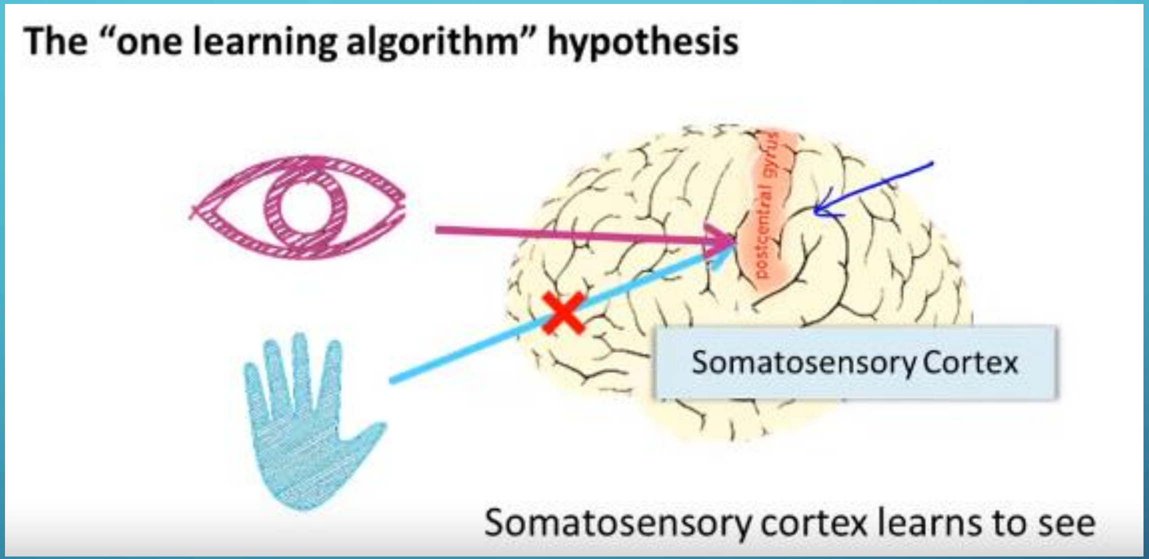
Origins: Algorithms that try to mimic the brain.
Was very widely used in 80s and early 90s; popularity diminished in late 90s.
Recent resurgence: State-of-the-art technique for many applications

OBSERVATION

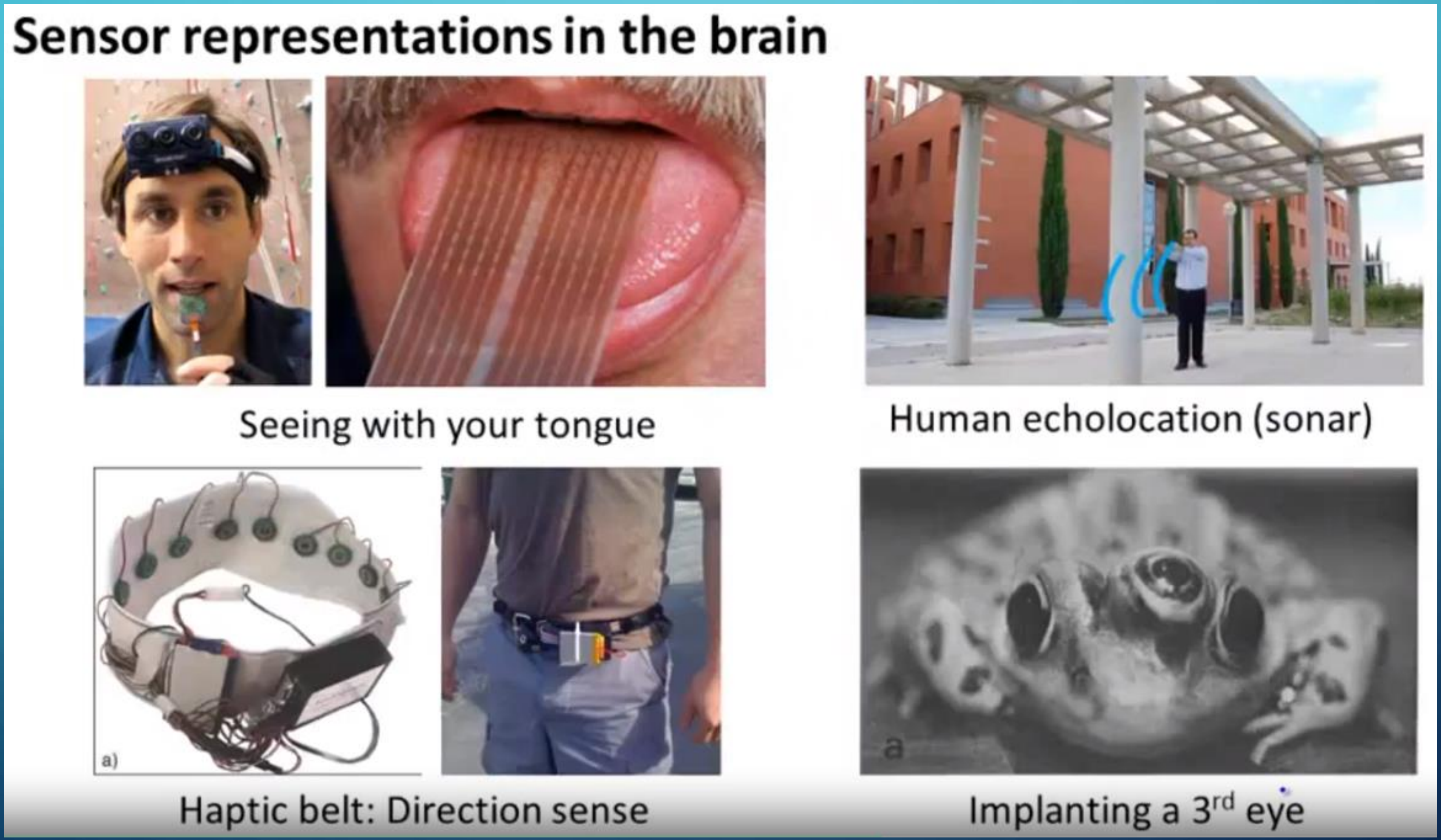



The “one learning algorithm” hypothesis
Auditory Cortex
Auditory cortex learns to see

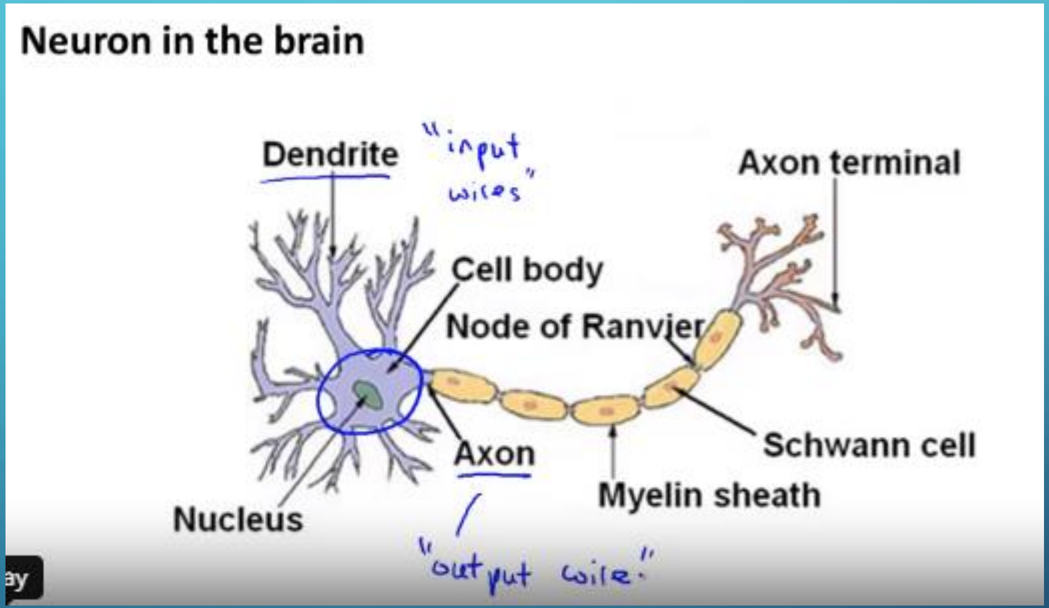
SIMILARLY


The "one learning algorithm" hypothesis
postcentral gyrus
Somatosensory Cortex
Somatosensory cortex learns to see

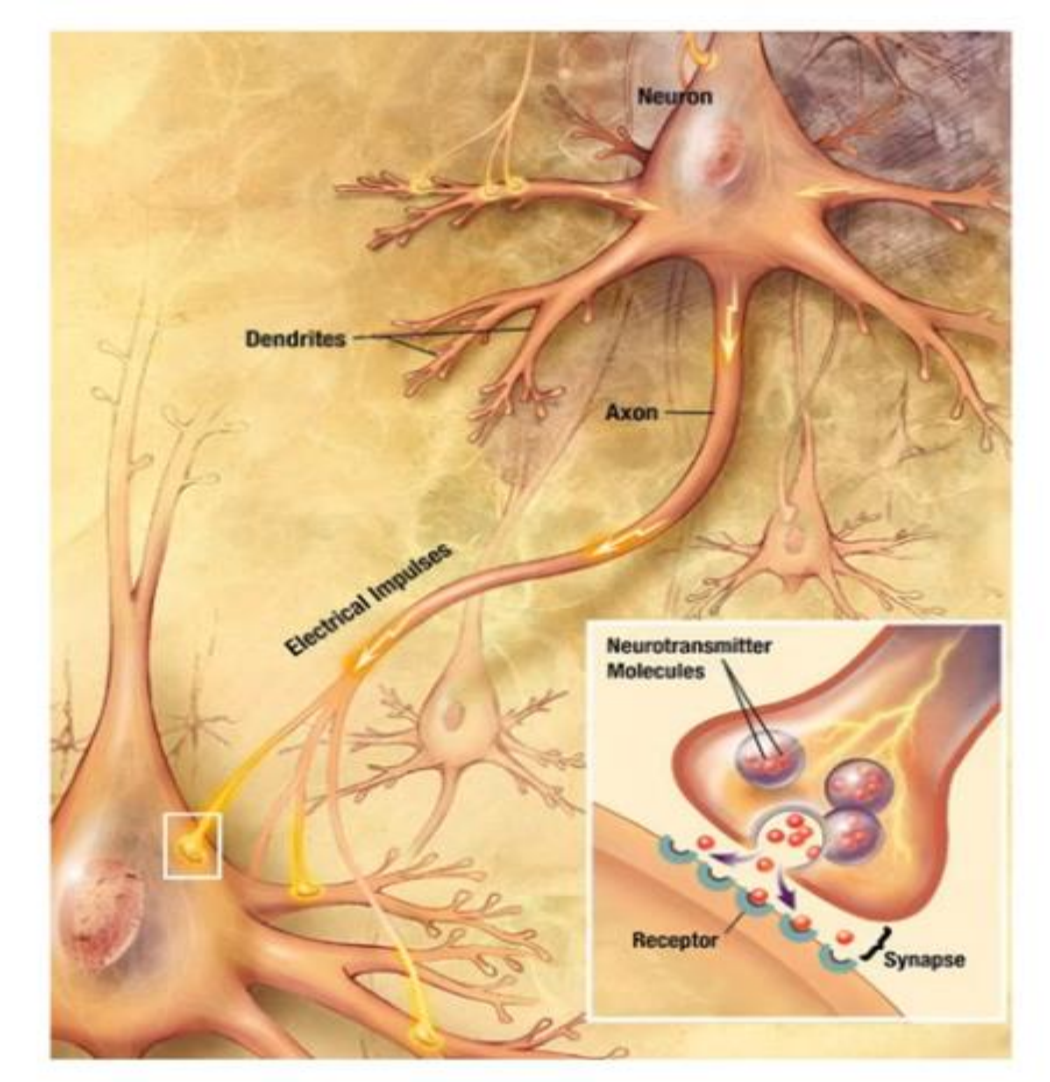
OTHER OBSERVATIONS



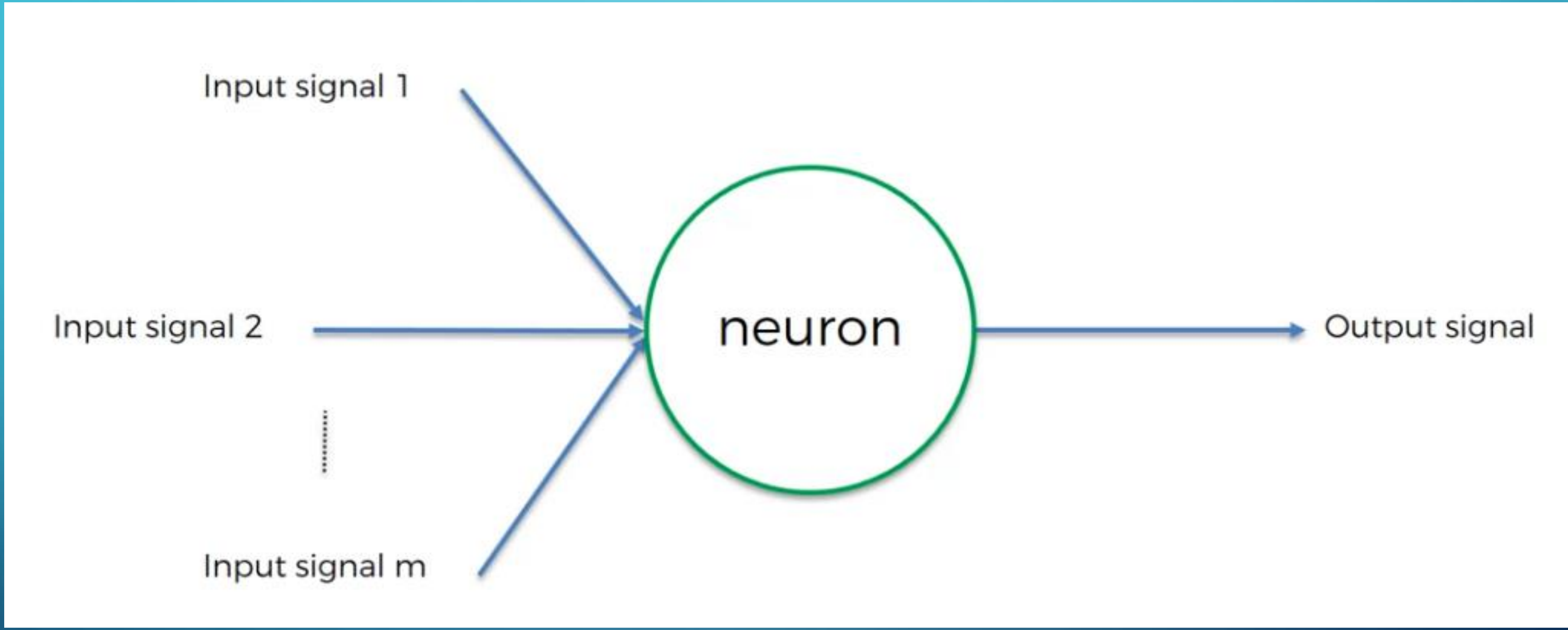
INSPIRATION FROM BRAIN



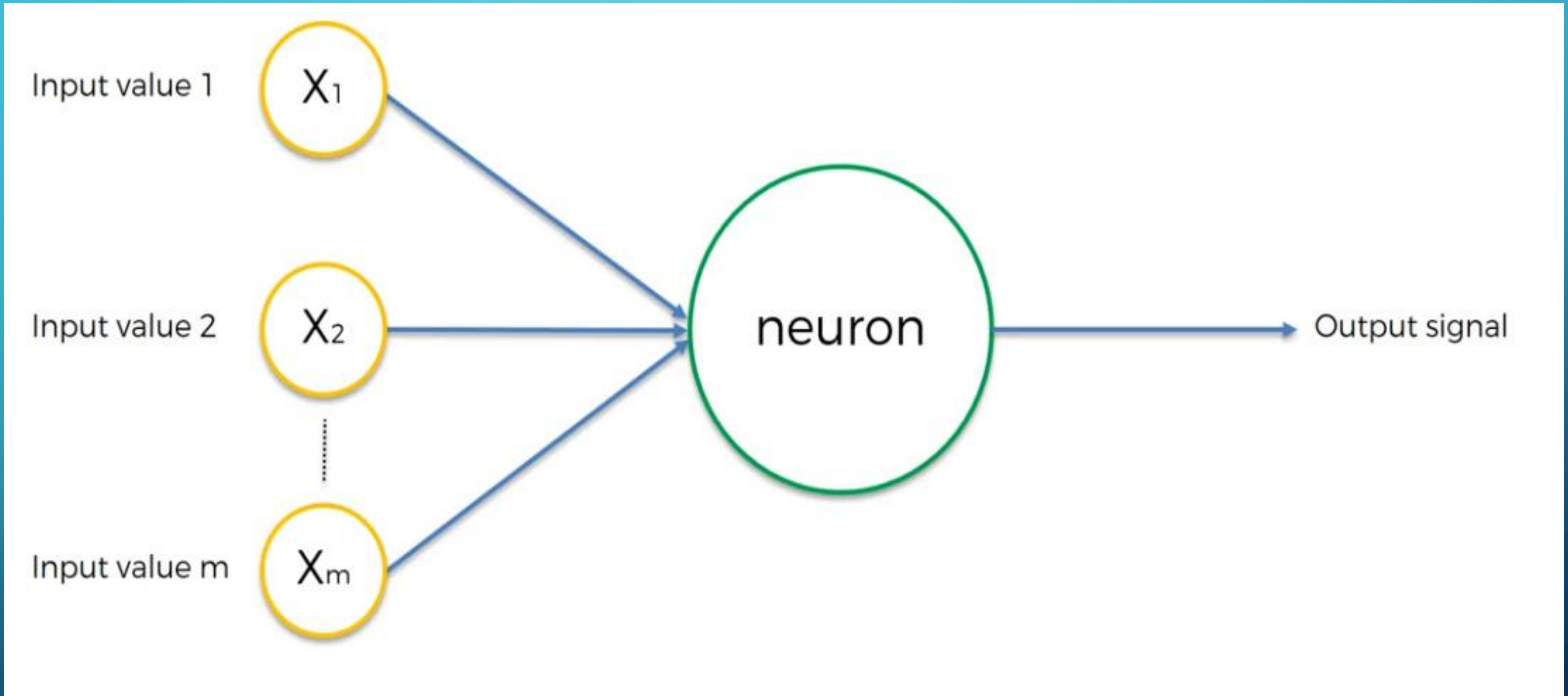
NEURON



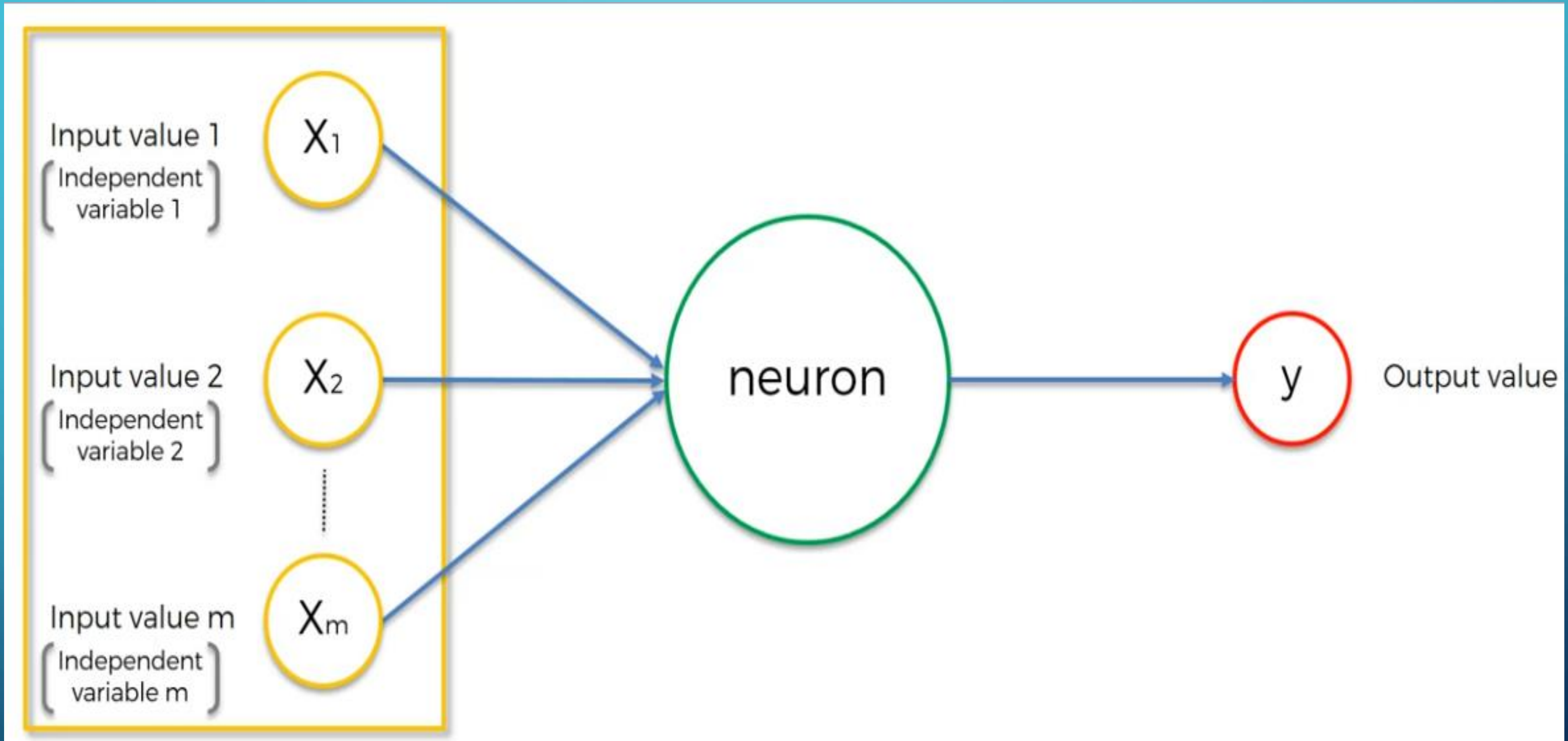
Neuron
Dendrites
Axon
Electrical Impulses
Neurotransmitter
Molecules
Receptor
Synapse

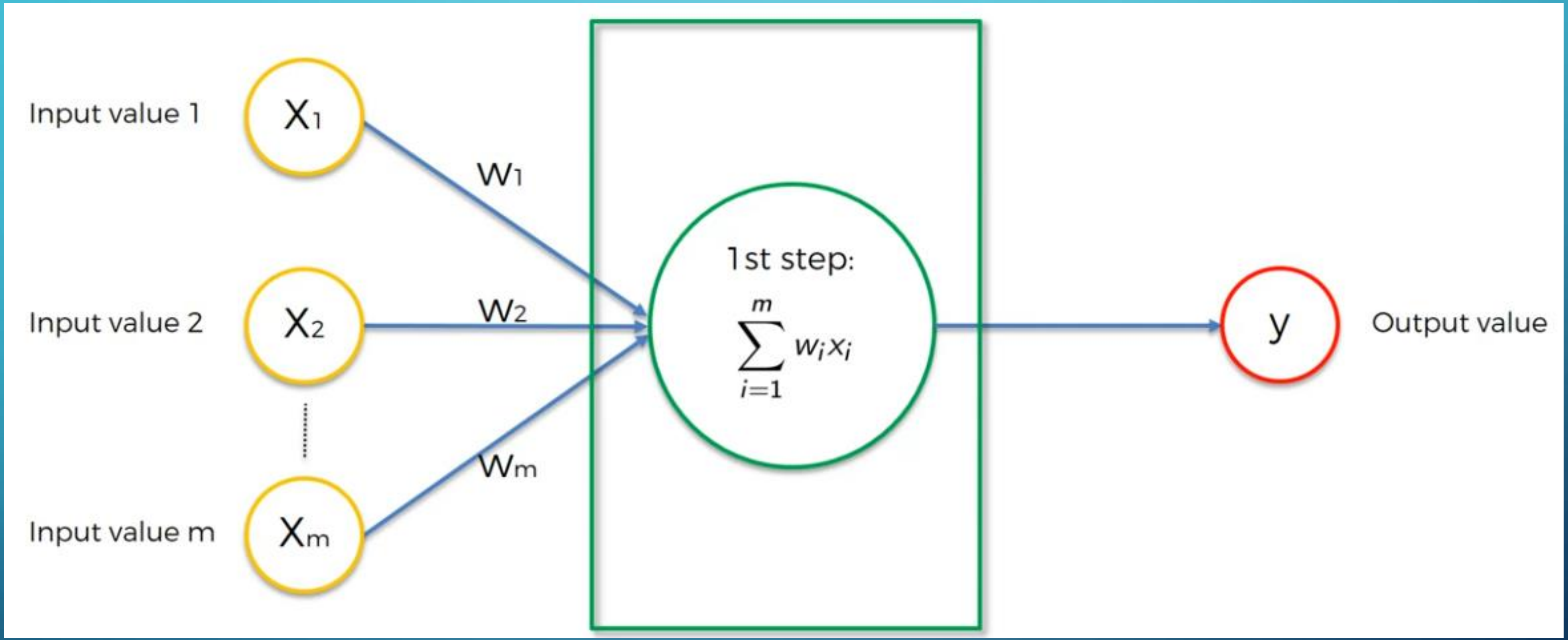

Input signal 1
Input signal 2
Input signal m
neuron
Output signal

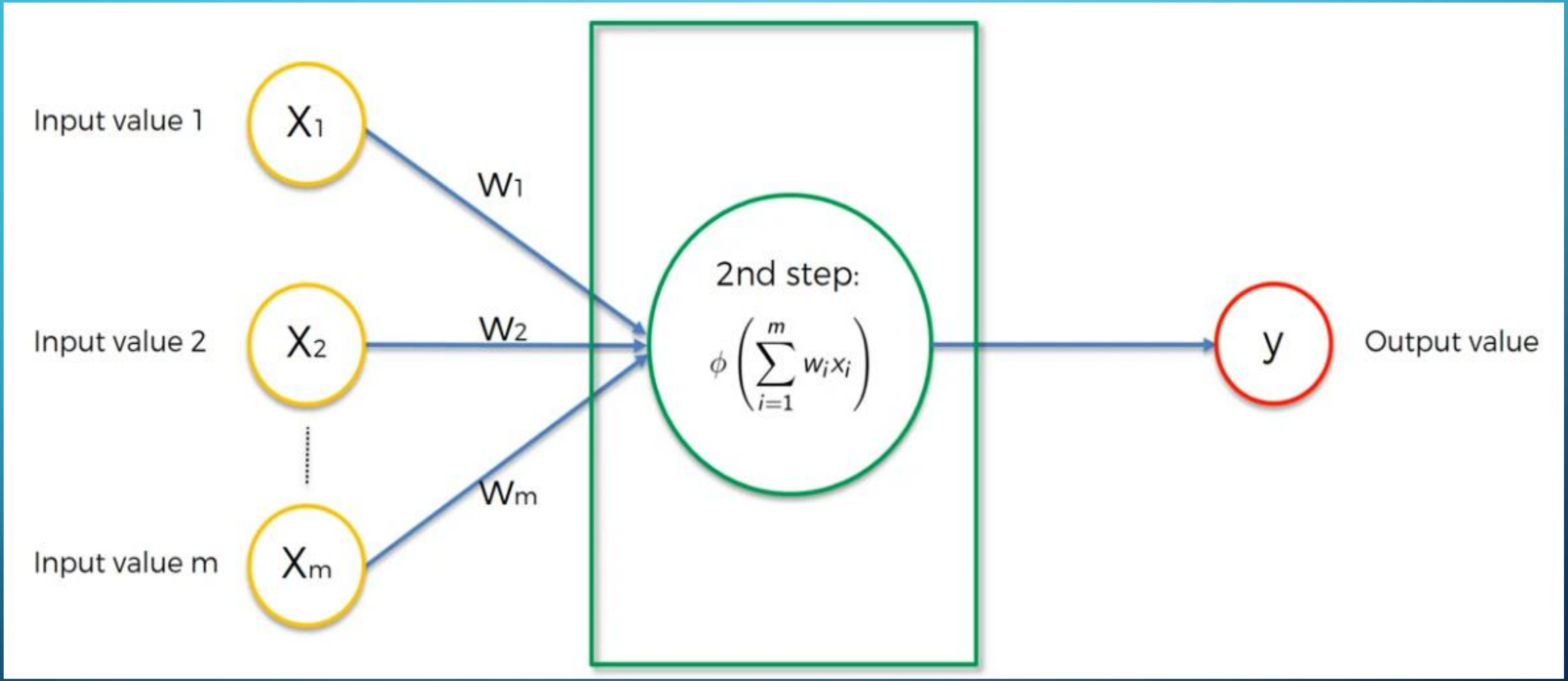
CAN BE REPRESENTED AS:

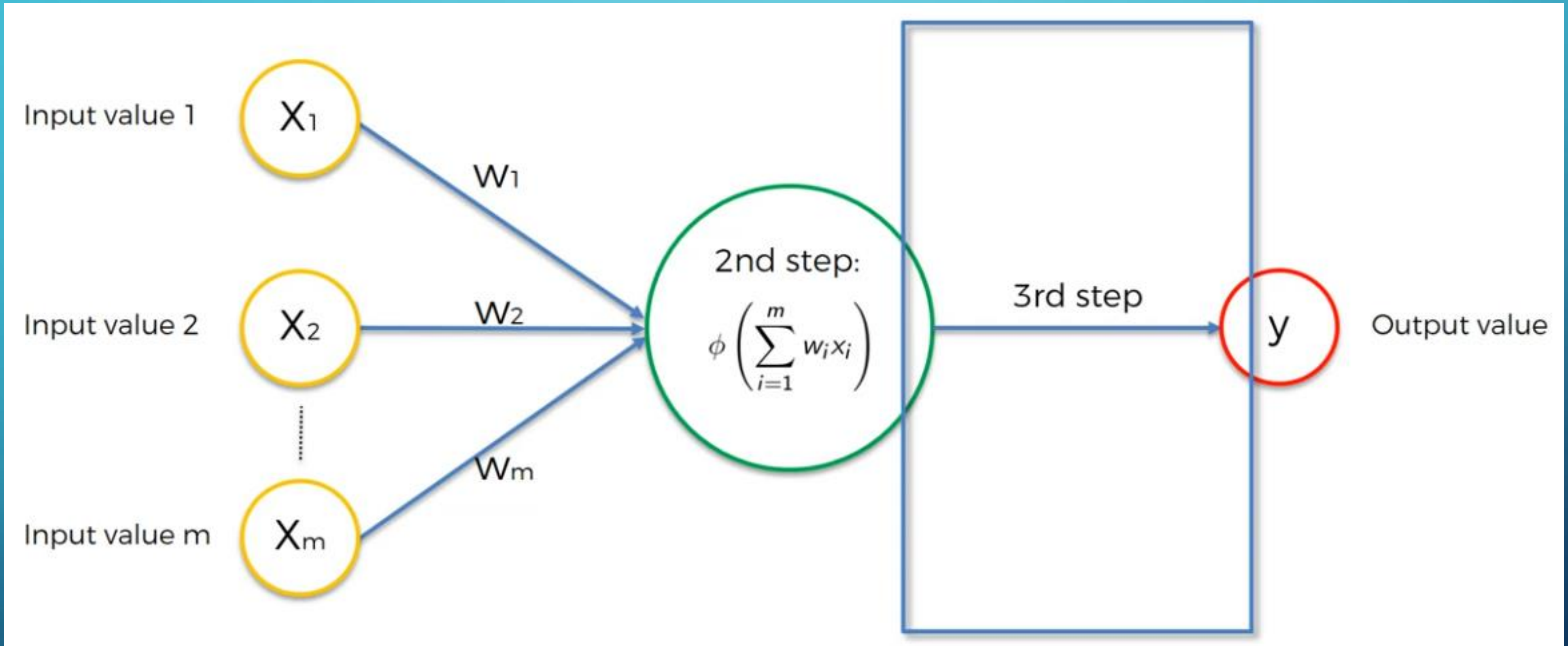


Similar to logistic or linear regression

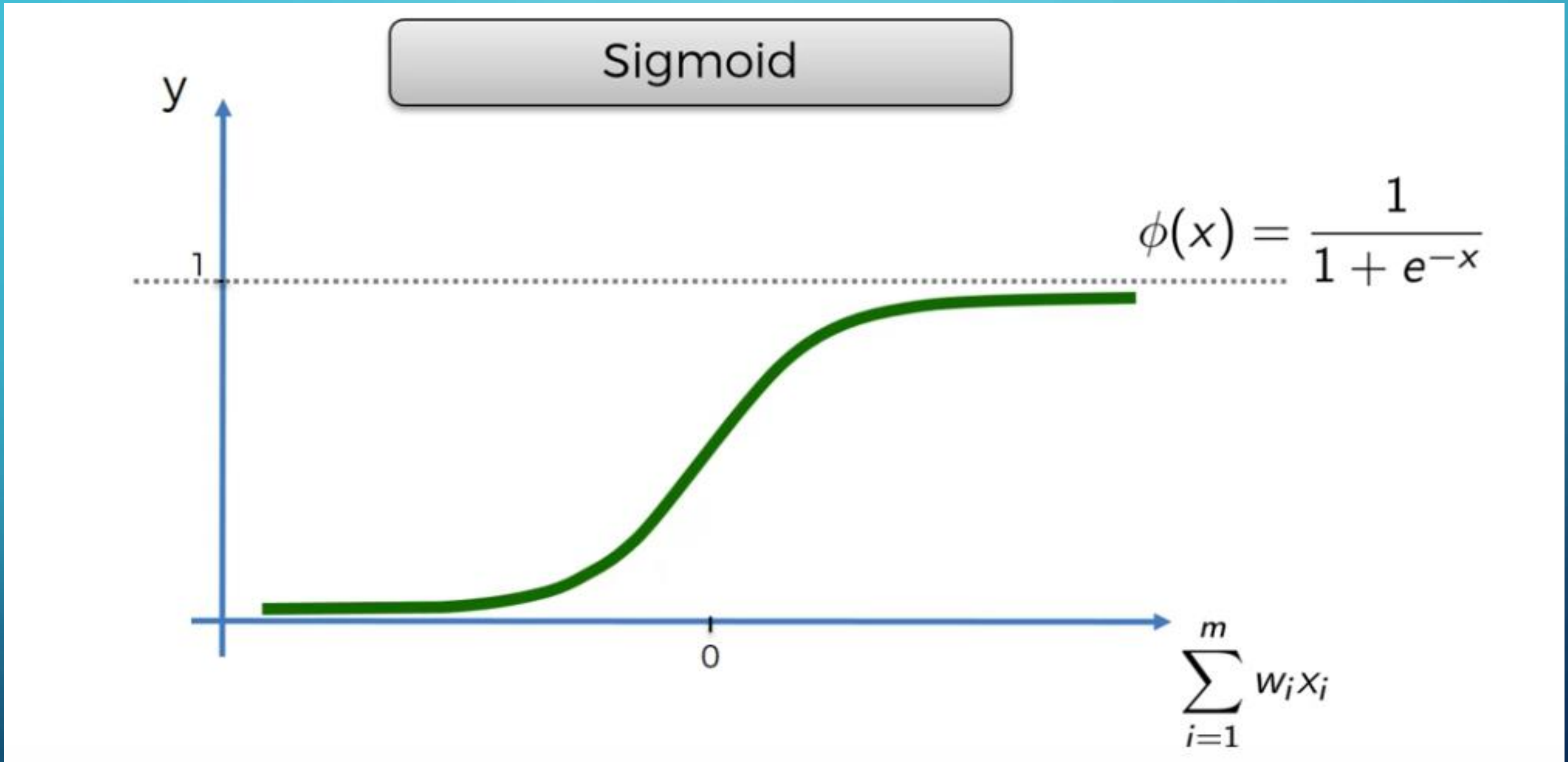

Input value 1
Independent variable 1
X1
Input value 2
Independent variable 2
X2
Input value m
Independent variable m
Xm
neuron
y
Output value

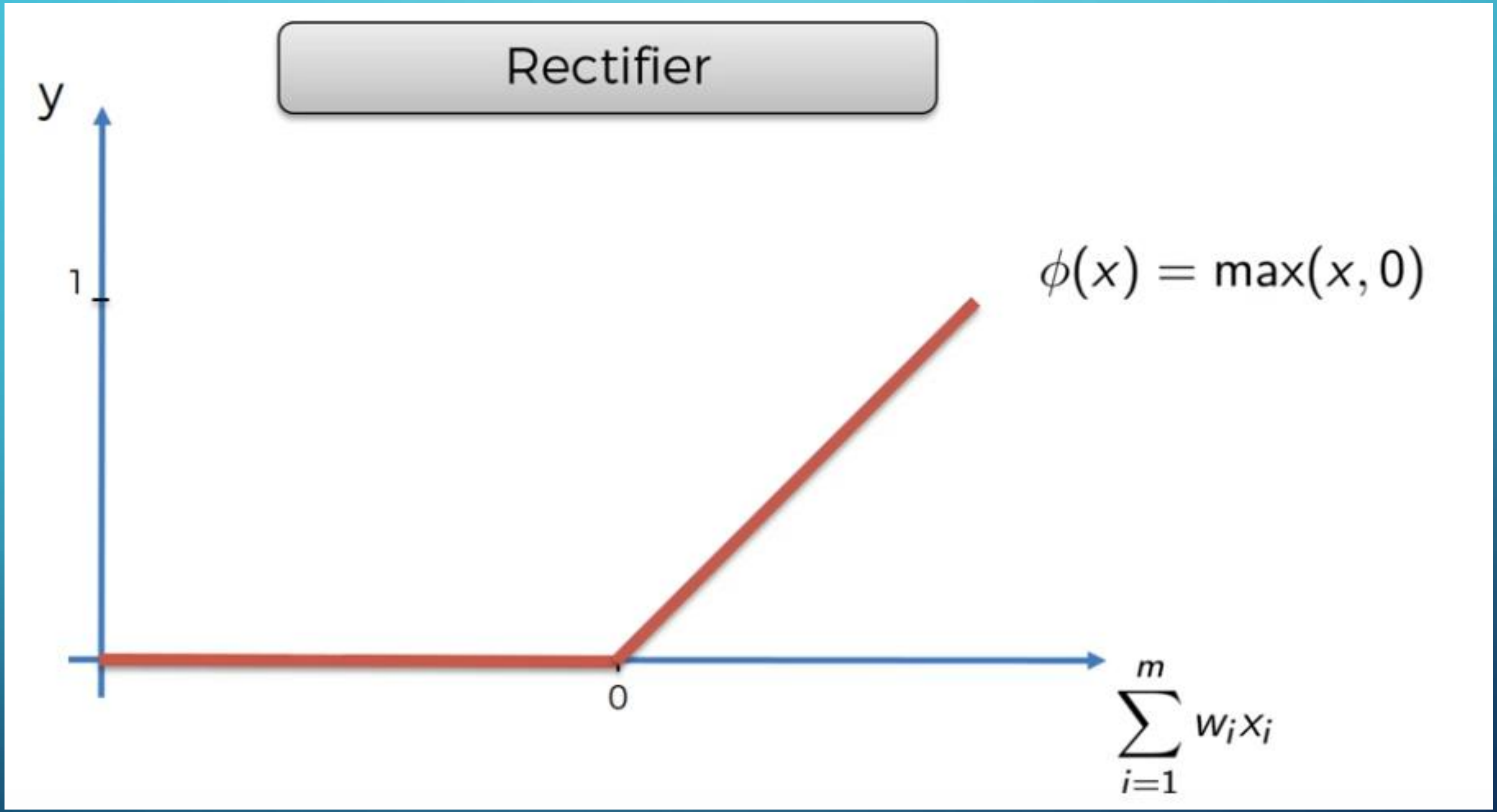

Input value 1
Input value 2
Input value m
X1
X2
Xm
W1
W2
Wm
1st step:
$\sum_{i=1}^{m} w_i x_i$
y
Output value

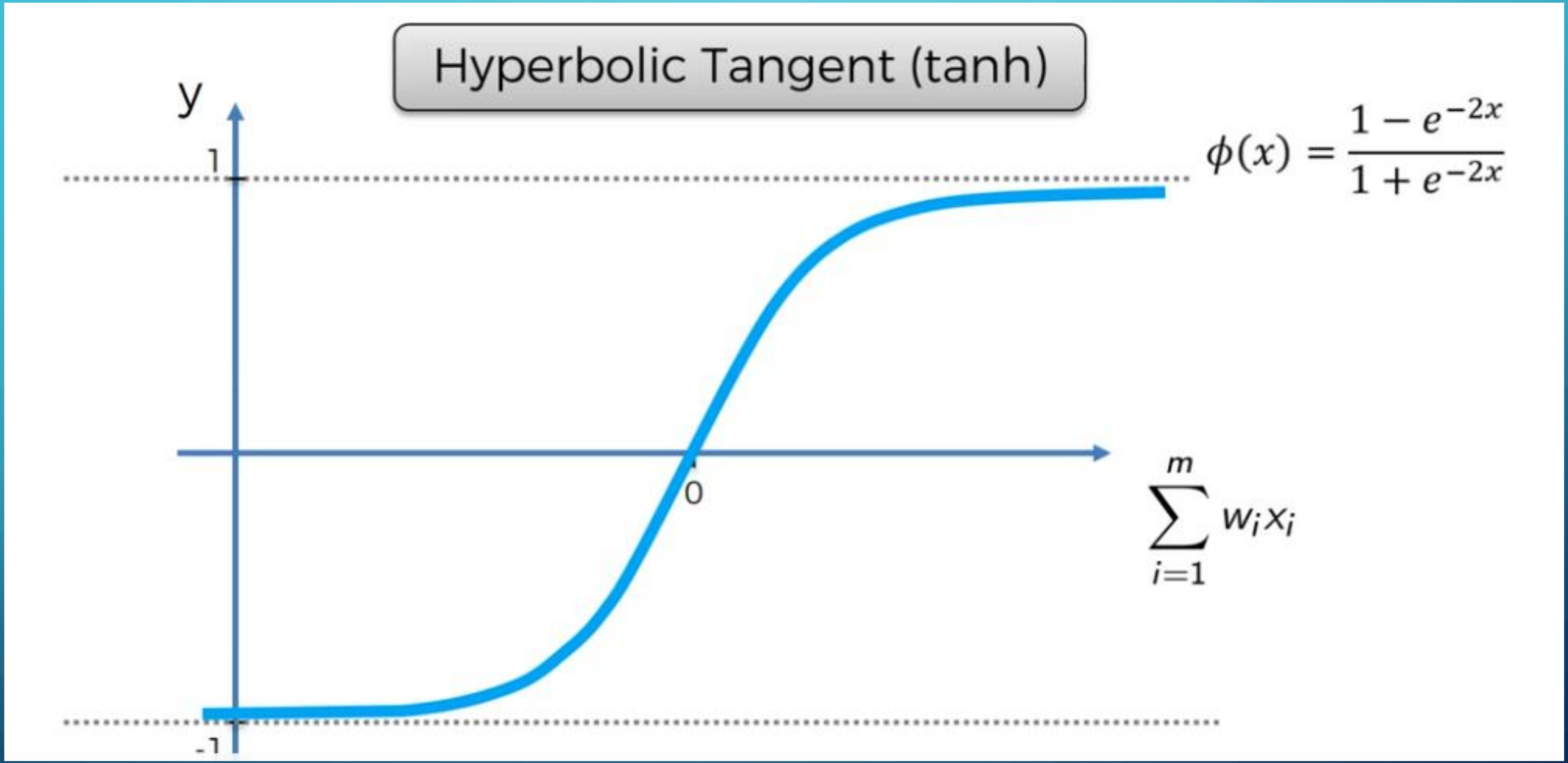

Input value 1
Input value 2
Input value m
X_1
X_2
X_m
W_1
W_2
W_m
2nd step:
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
y
Output value


Input value 1
Input value 2
Input value m
X_1
X_2
X_m
W_1
W_2
W_m
2nd step:
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
3rd step
y
Output value

ACTIVATION FUNCTIONS


Sigmoid
y
1
0
$\phi(x) = \frac{1}{1 + e^{-x}}$
$\sum_{i=1}^{m} w_i x_i$

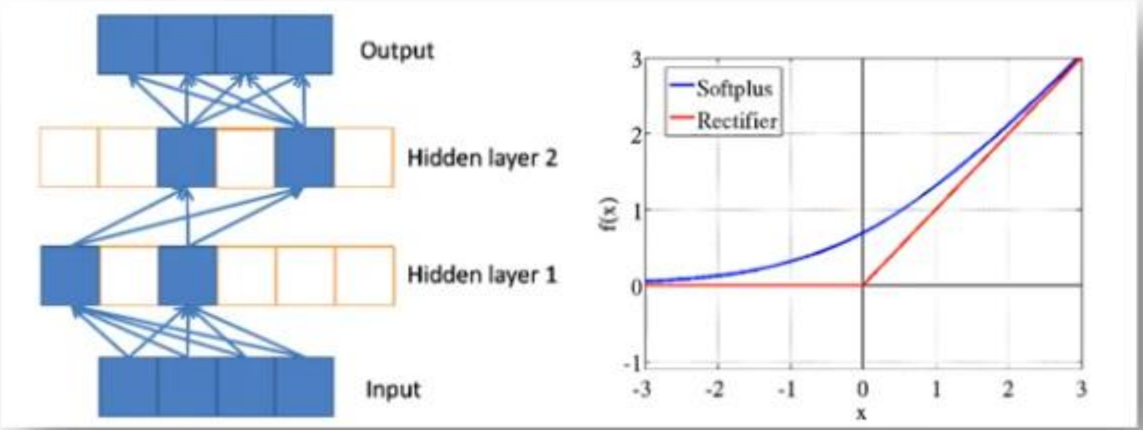

Rectifier
y
1
0
$\phi(x) = \max(x, 0)$
$\sum_{i=1}^{m} w_i x_i$


Hyperbolic Tangent (tanh)
y
1
-1
0
$\phi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$
$\sum_{i=1}^{m} w_i x_i$

Additional Reading:

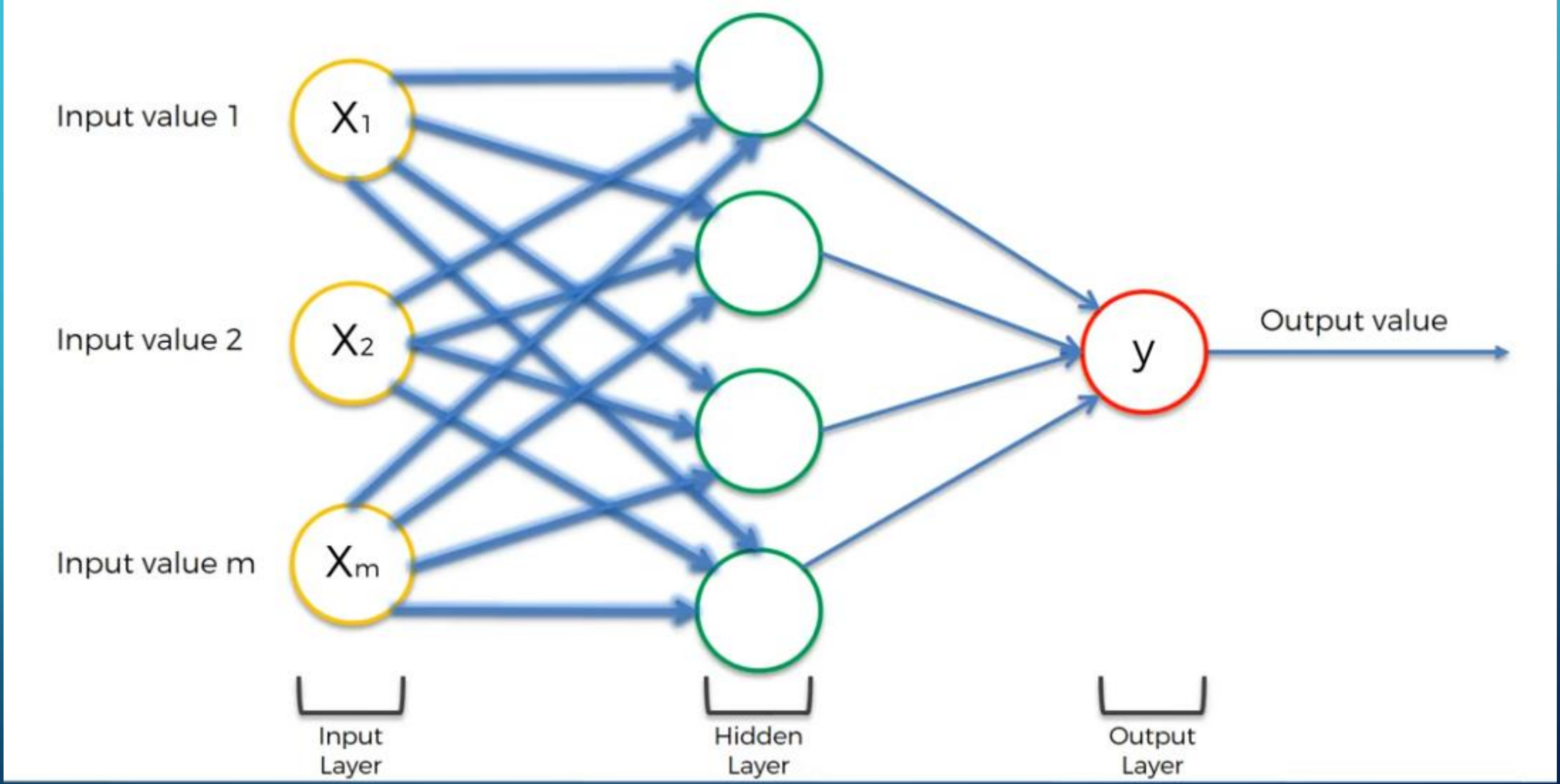
Deep sparse rectifier neural networks

By Xavier Glorot et al. (2011)

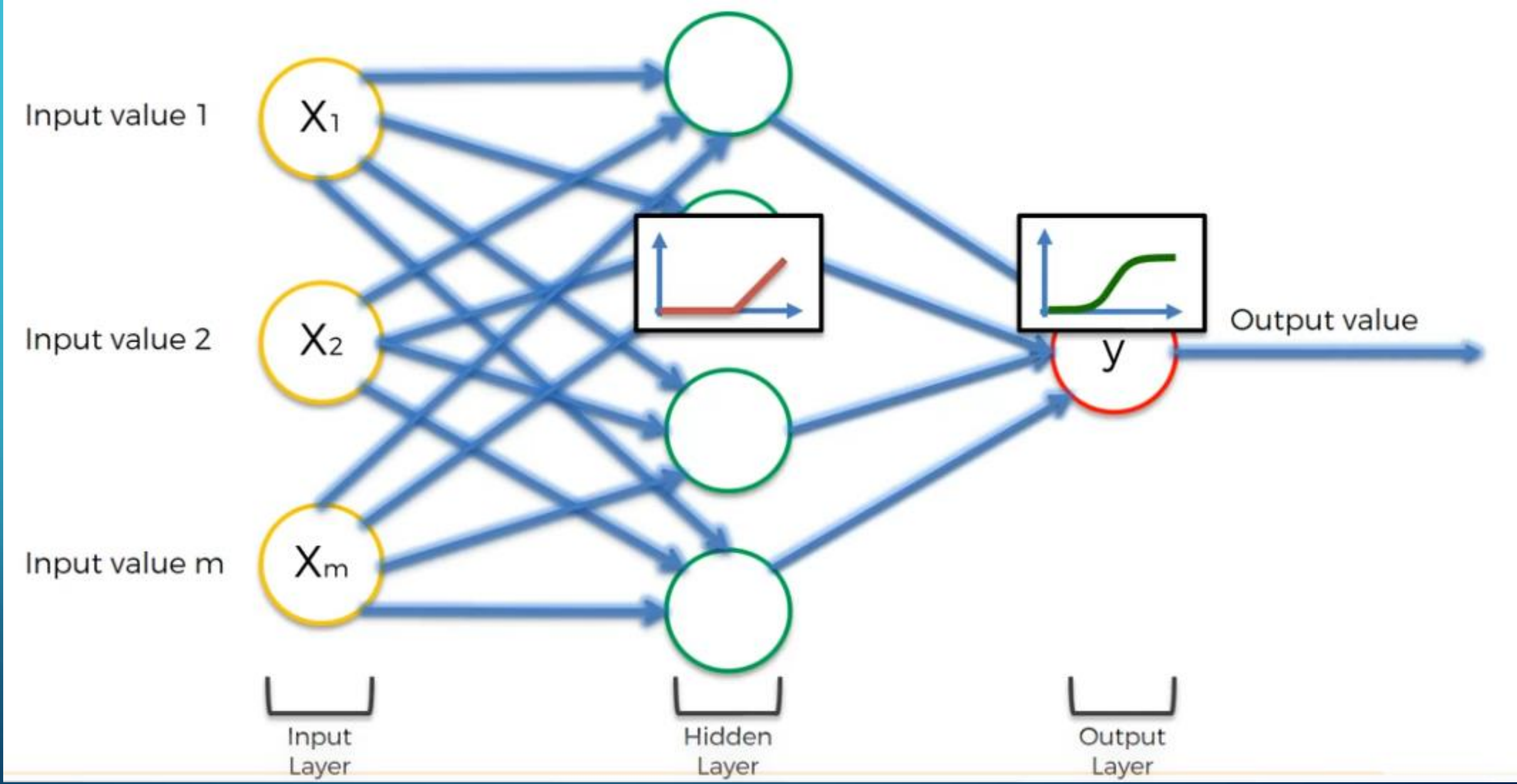


Link:

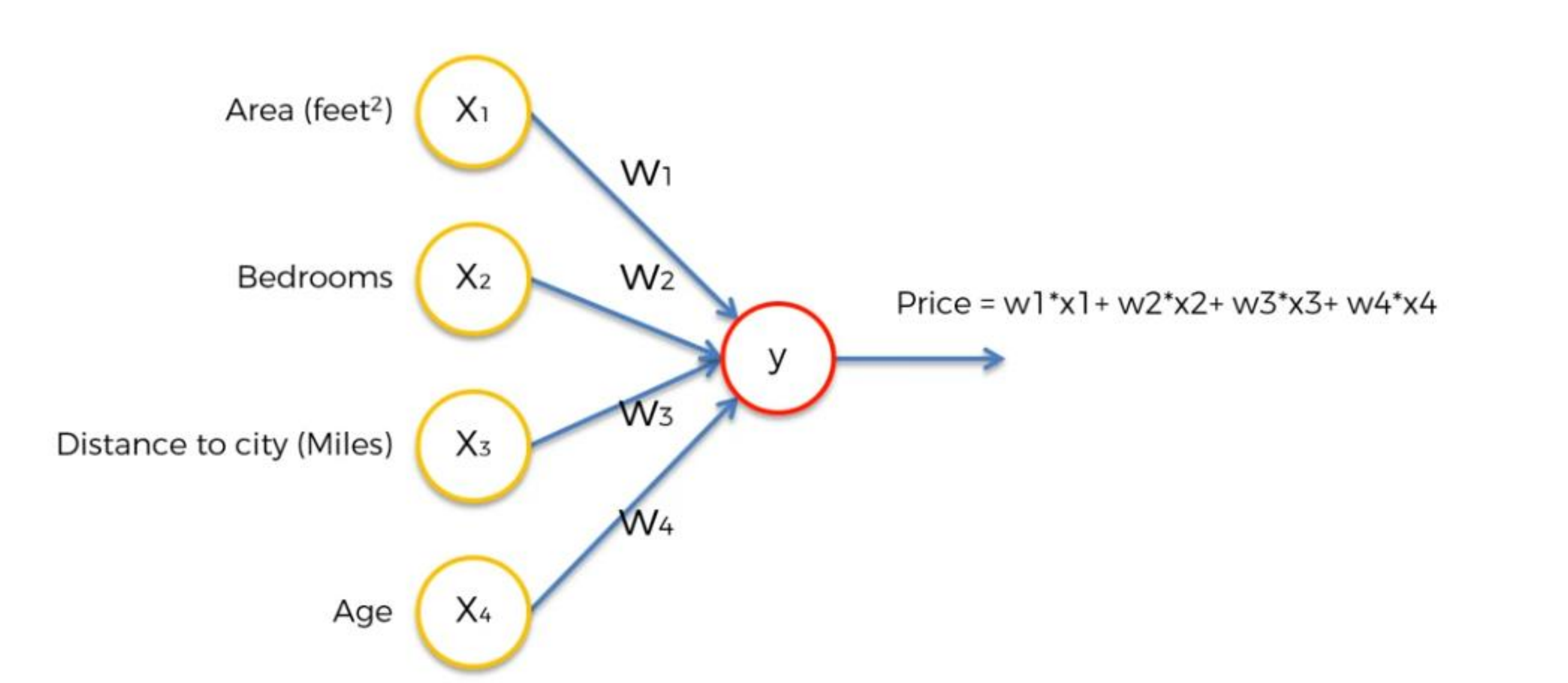
http://jmlr.org/proceedings/papers/v15/glorot11a/glorot11a.pdf



Input value 1
Input value 2
Input value m
X_1
X_2
X_m
y
Output value
Input Layer
Hidden Layer
Output Layer

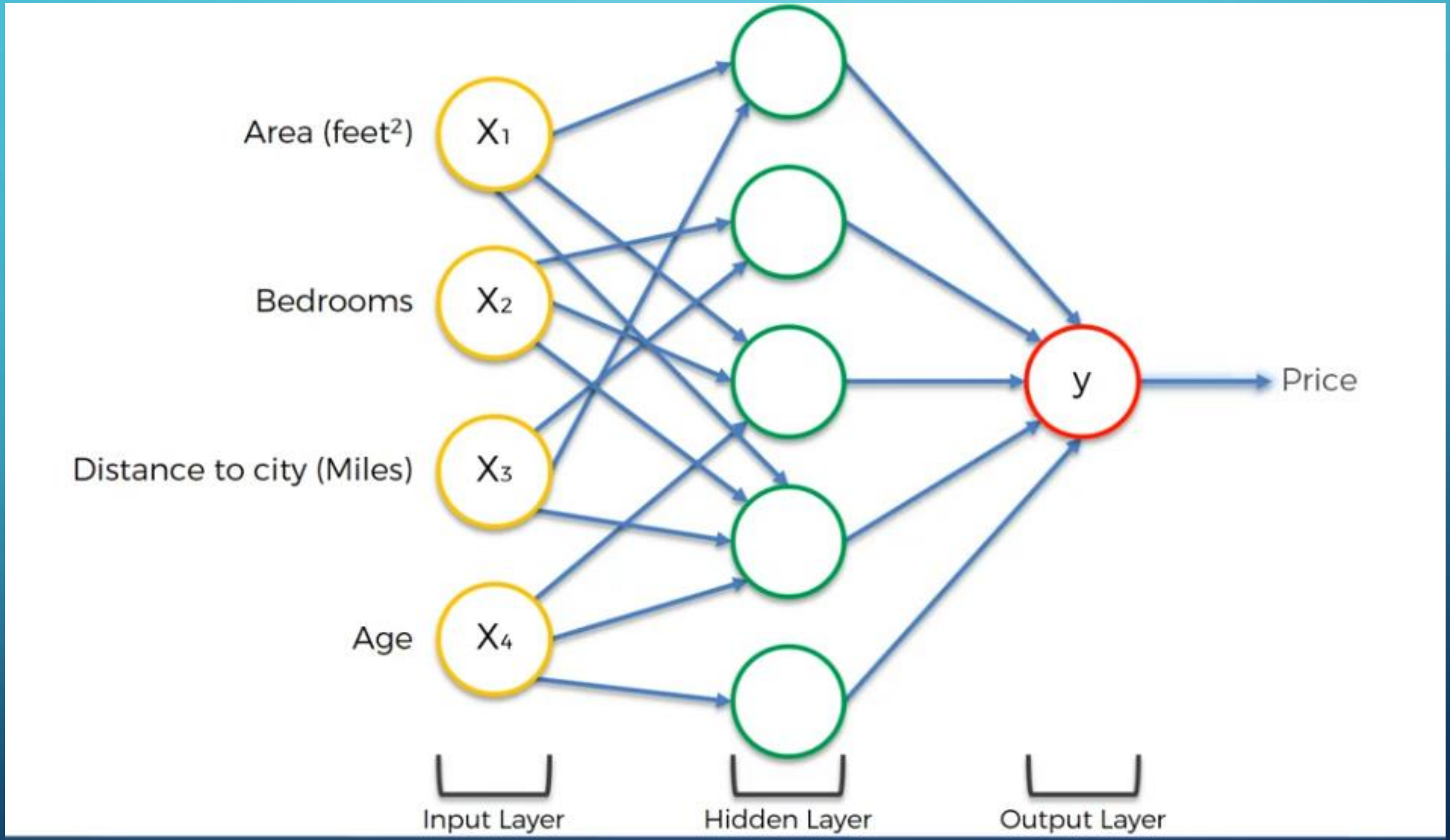

Input value 1
Input value 2
Input value m
X_1
X_2
X_m
y
Output value
Input Layer
Hidden Layer
Output Layer

REVISITING THE HOUSING PRICE EXAMPLE

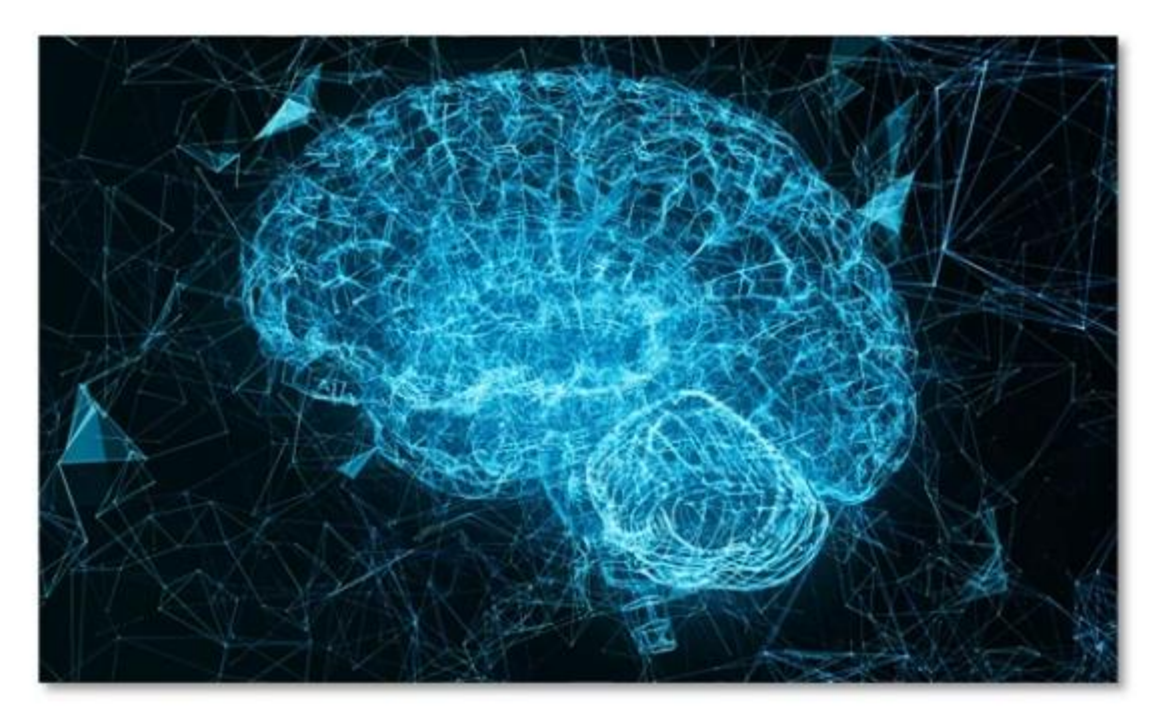

Area (feet²)
X1
Bedrooms
X2
Distance to city (Miles)
X3
Age
X4
W1
W2
W3
W4
y
Price = w1*x1+ w2*x2+ w3*x3+ w4*x4

PROBLEM OF NON-LINEARITY WITH THIS MODEL

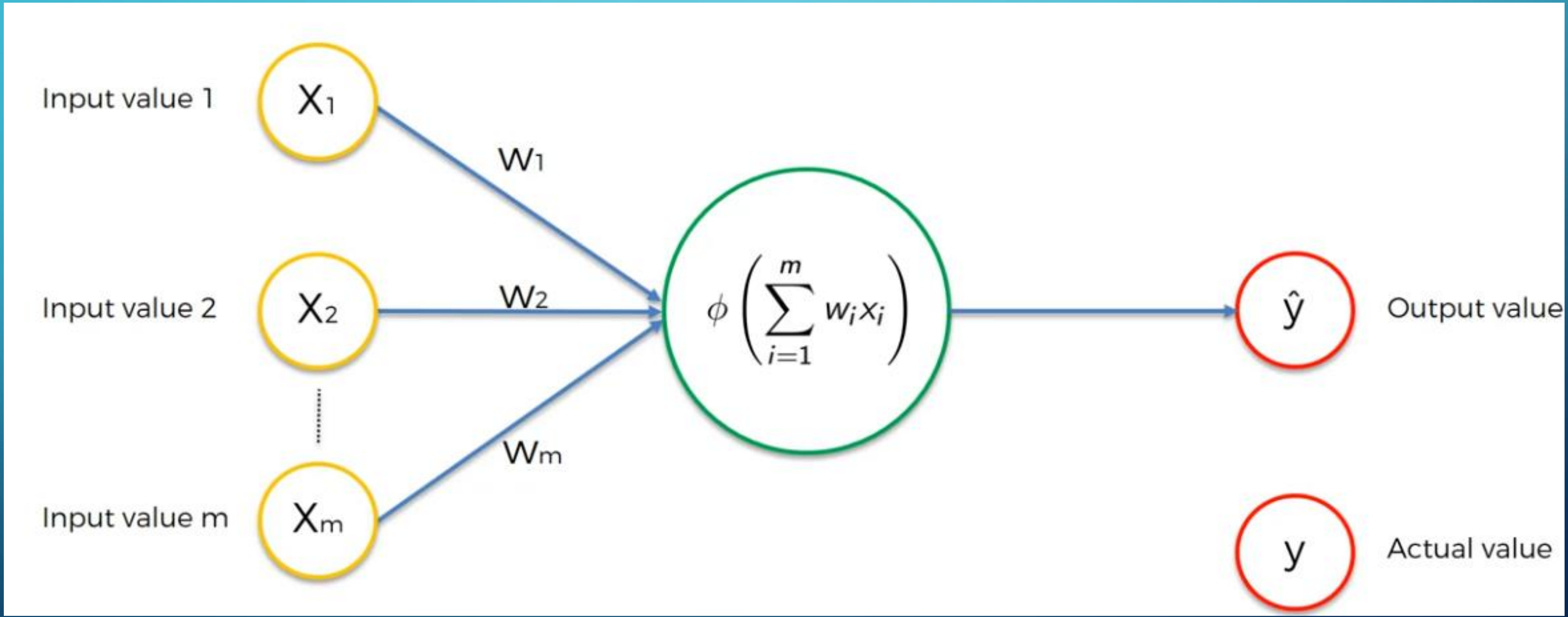
ADDING A HIDDEN LAYER

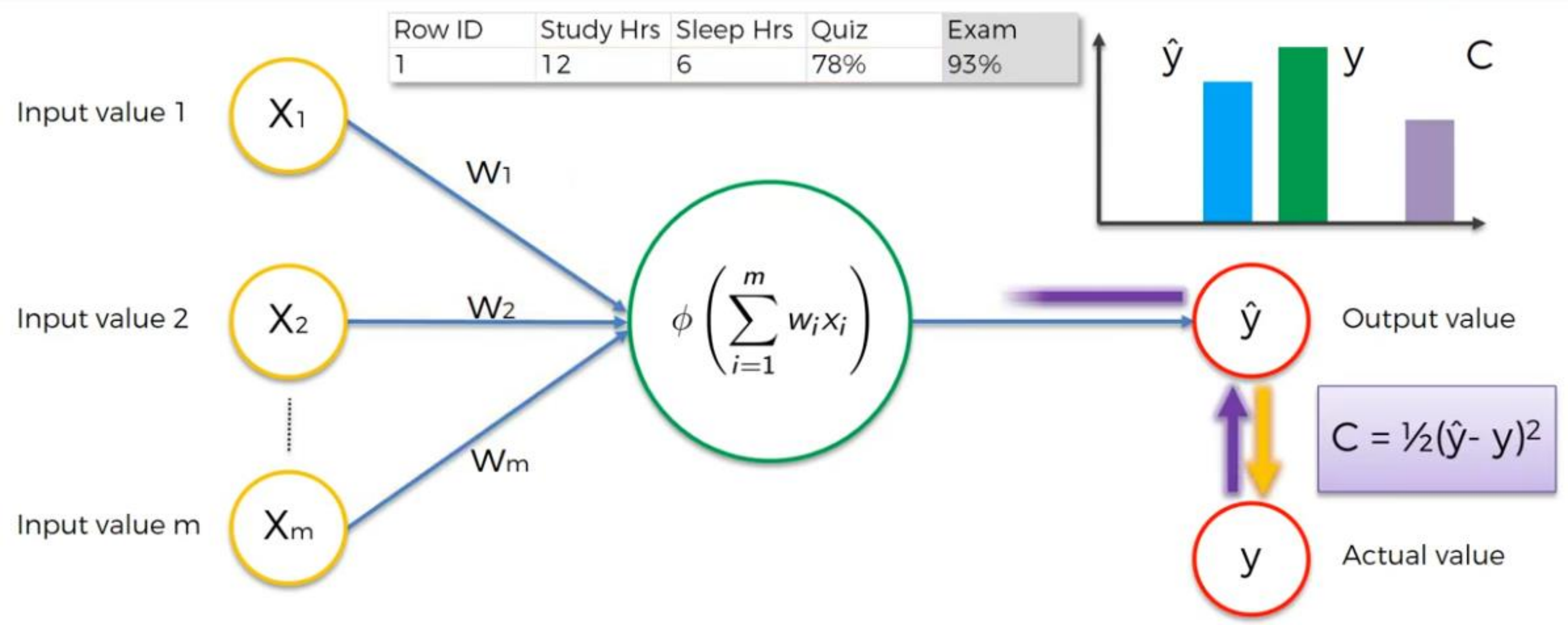


REMEMBER HOW YOU LEARNED ORANGES AND APPLES



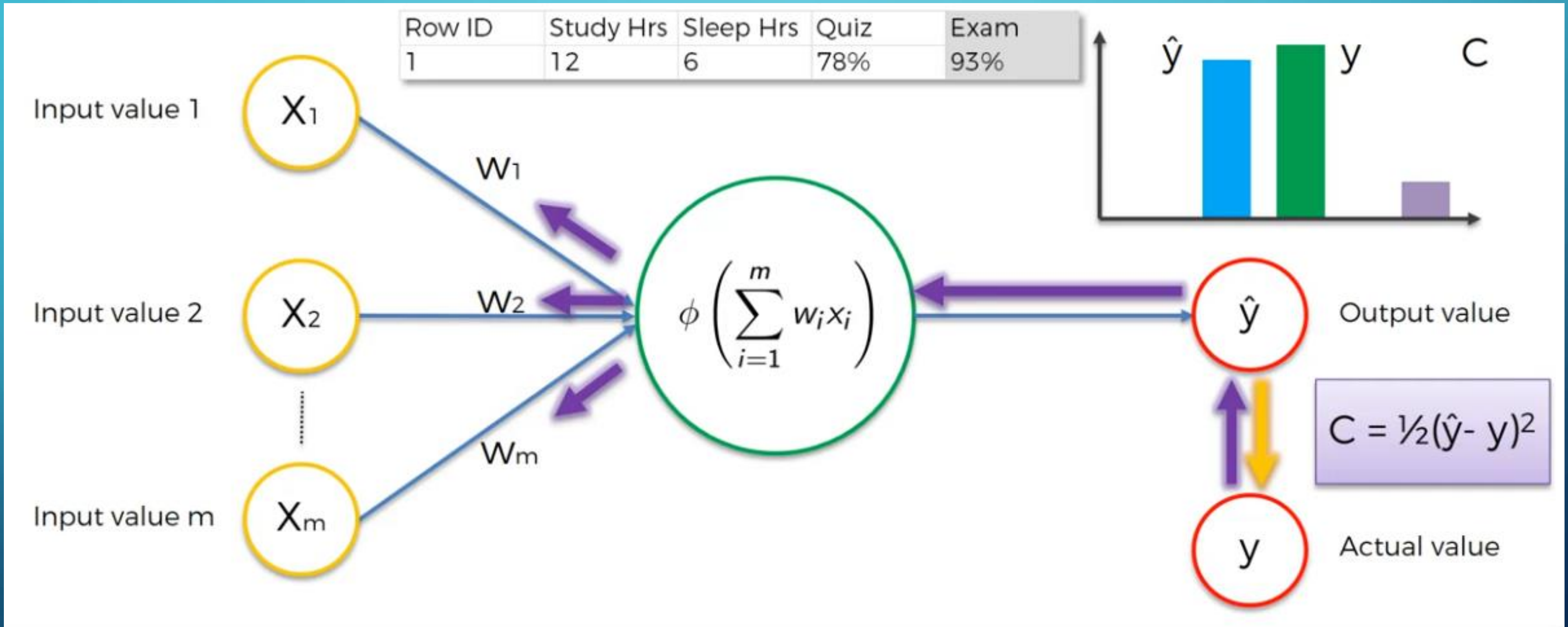
IN LOGISTIC REGRESSION



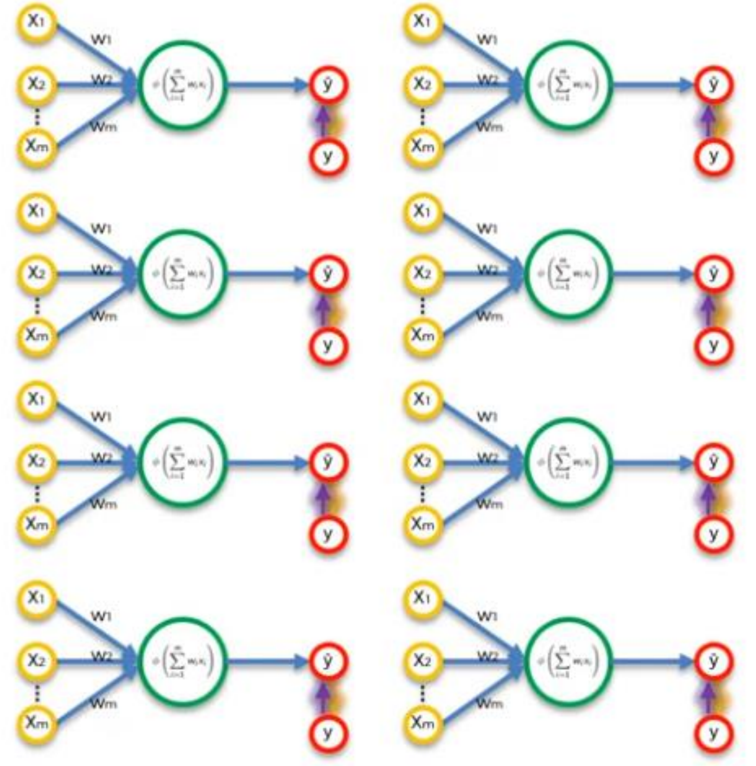


Remember Linear Regression

CALCULATING GRADIENTS USING LOSS



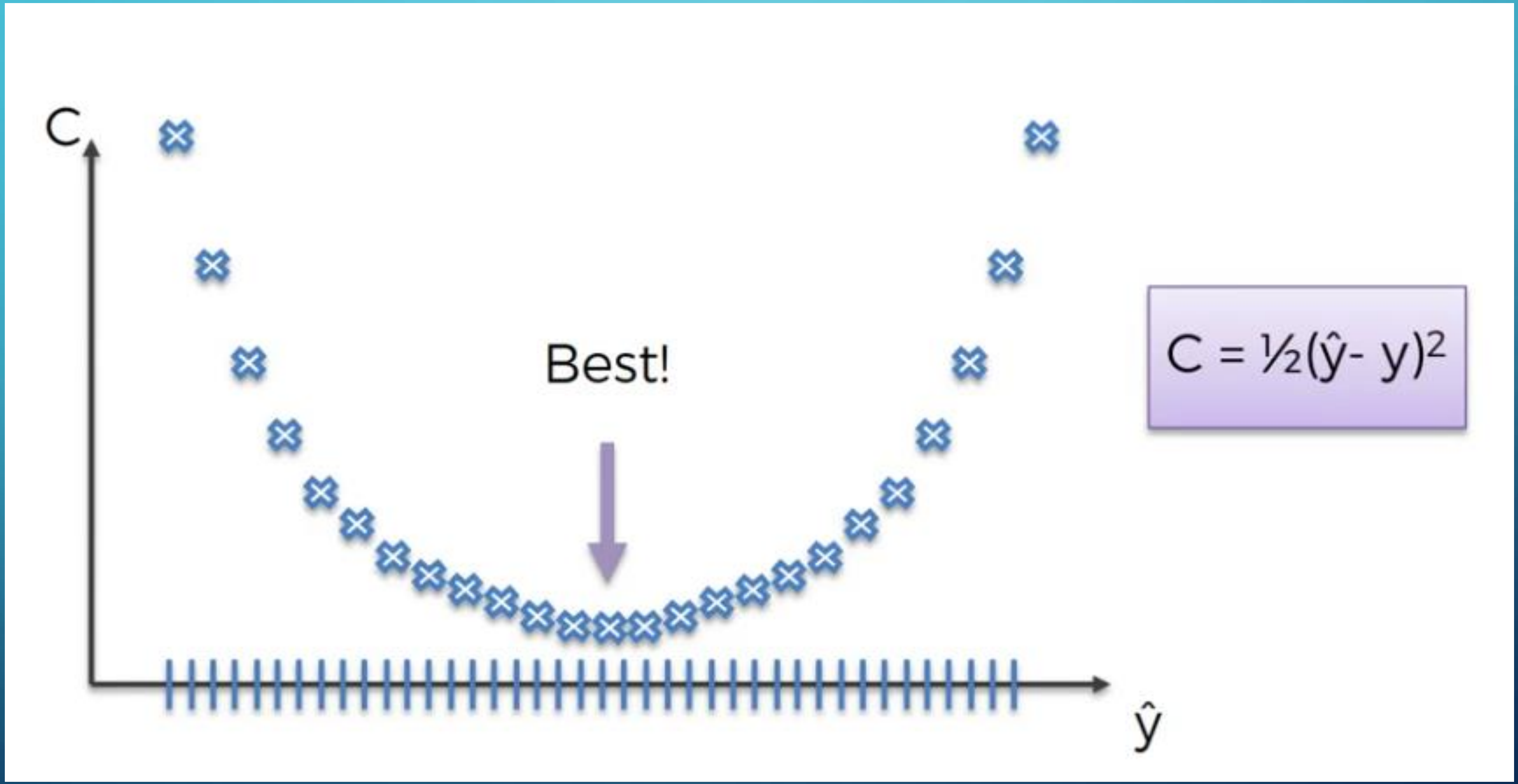
SUMMARISING LINEAR REGRESSION

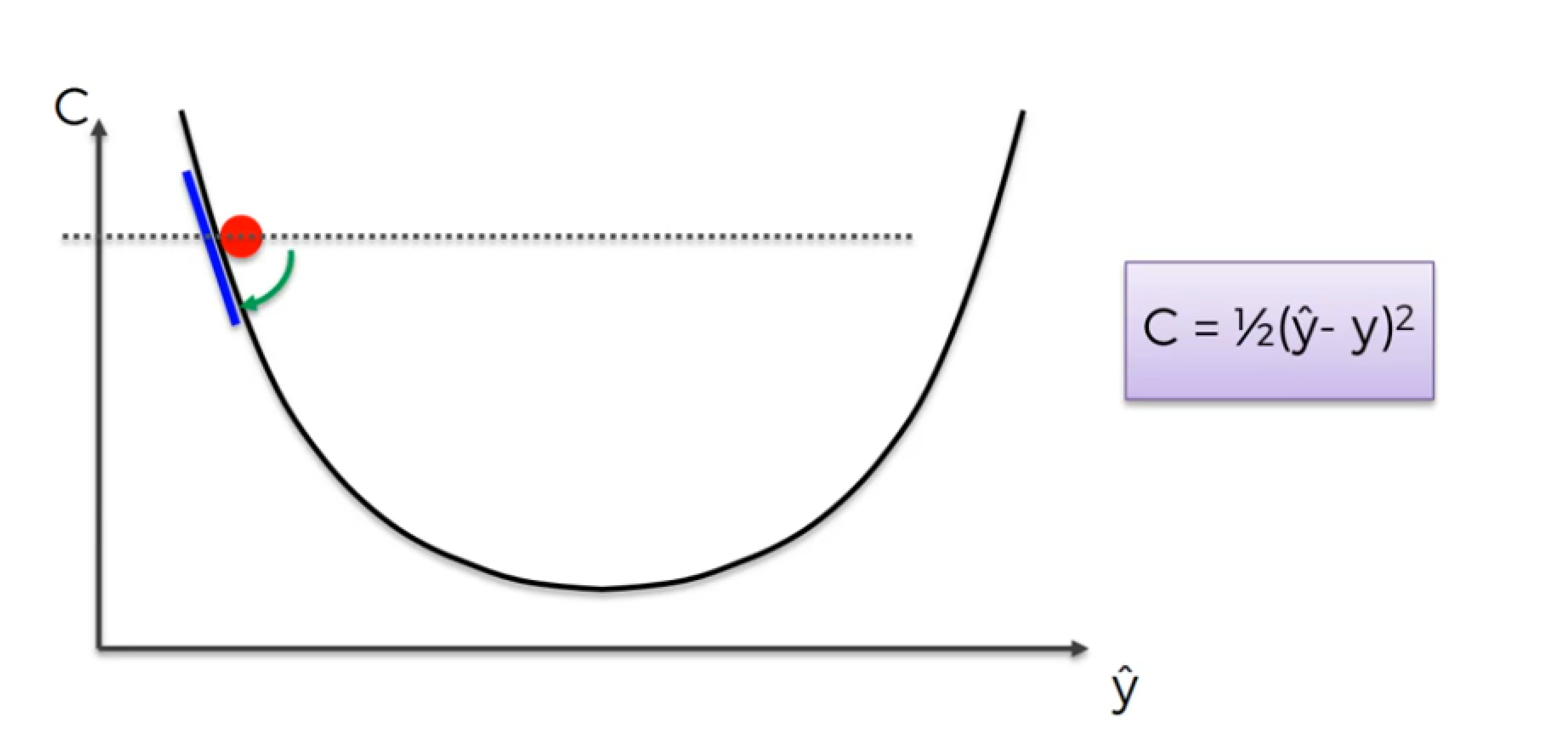


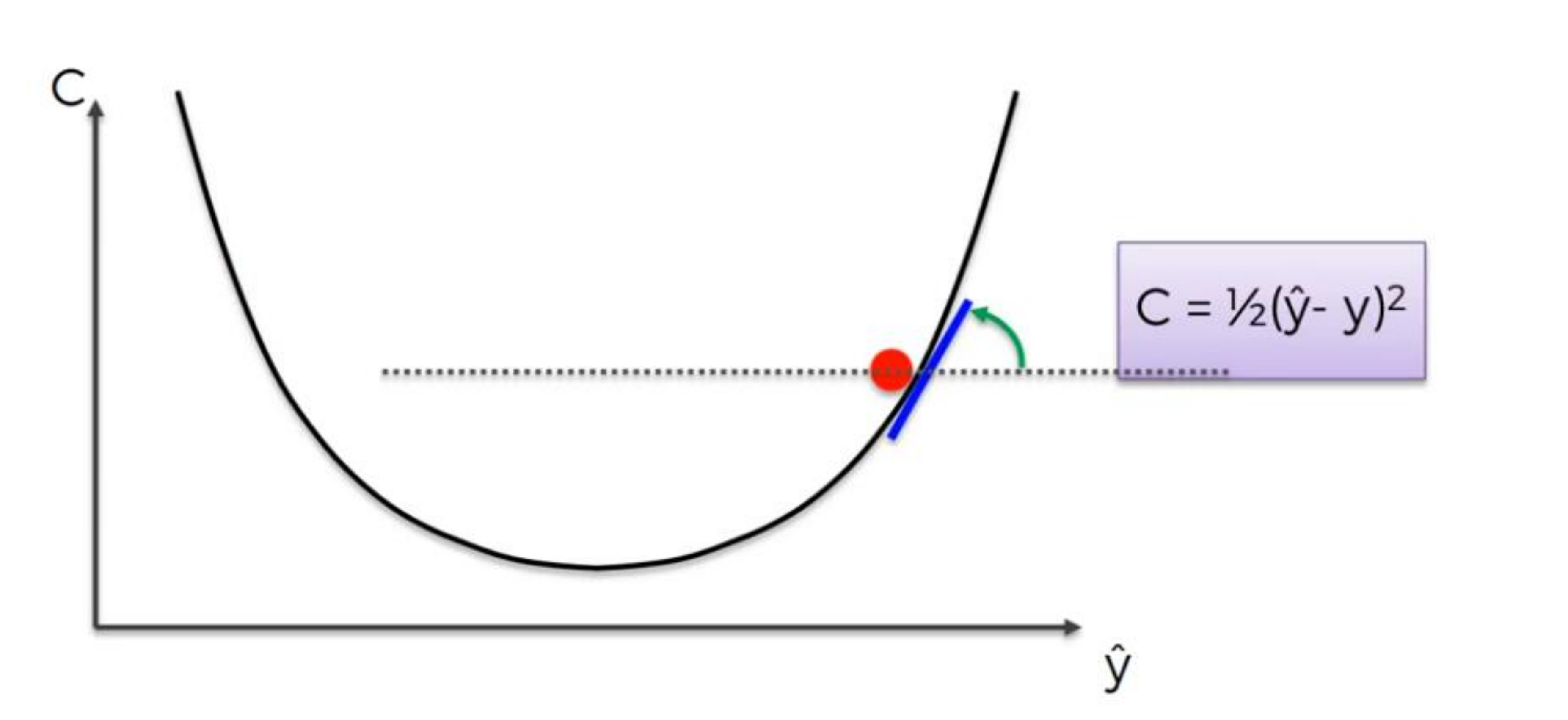
Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
1	12	6	78%	93%
2	22	6.5	24%	68%
3	115	4	100%	95%
4	31	9	67%	75%
5	0	10	58%	51%
6	5	8	78%	60%
7	92	6	82%	89%
8	57	8	91%	97%

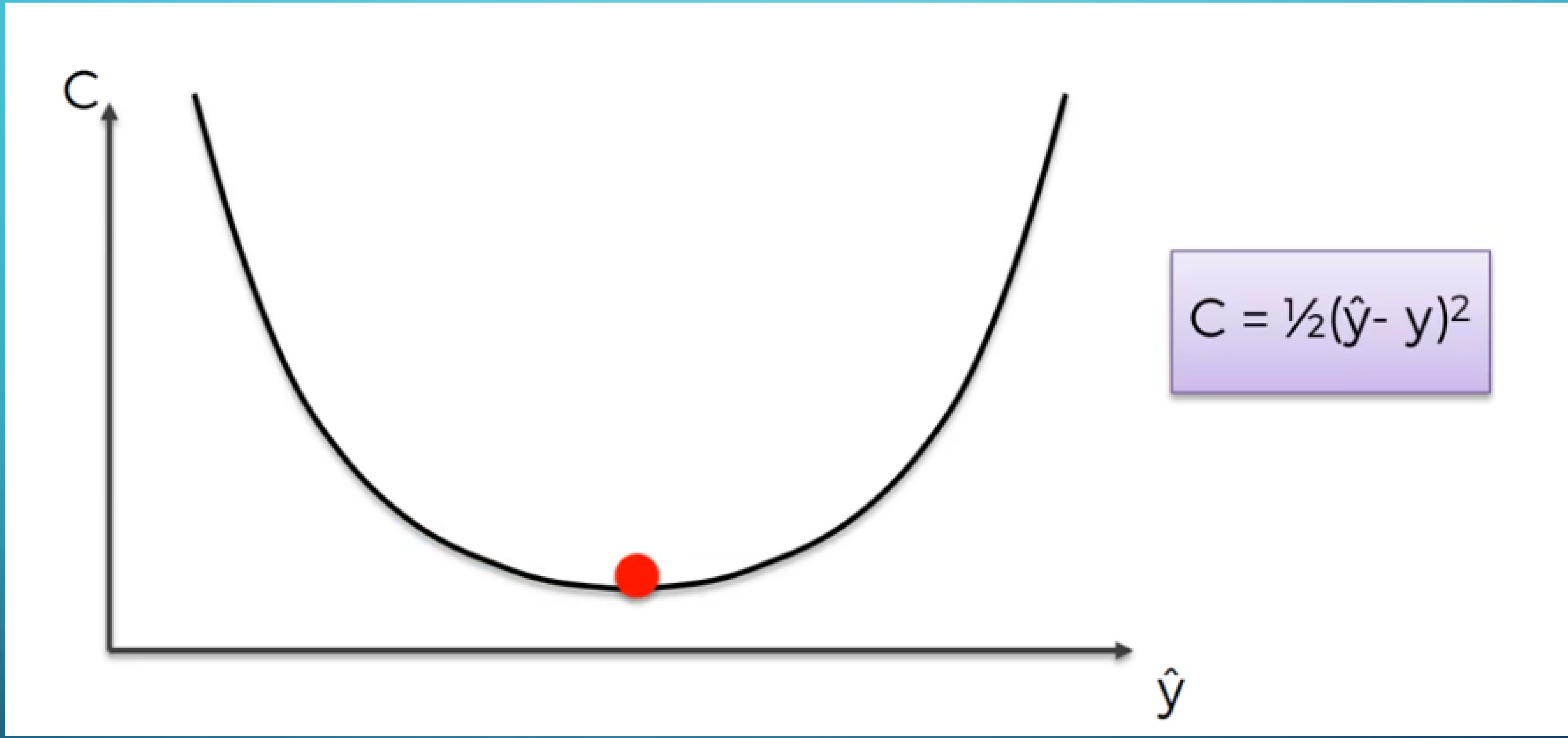
$C = \sum \frac{1}{2}(\hat{y} - y)^2$

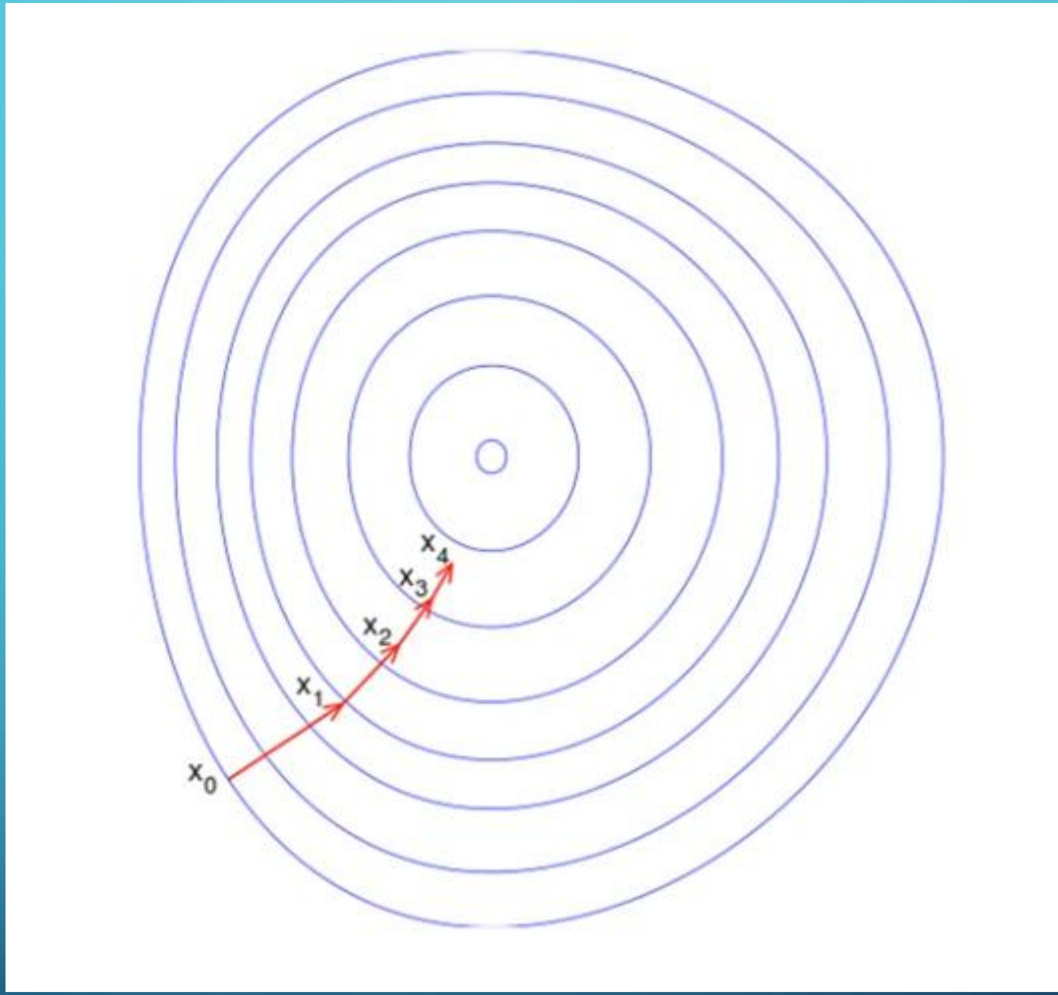
RECALL GRADIENT DESCENT

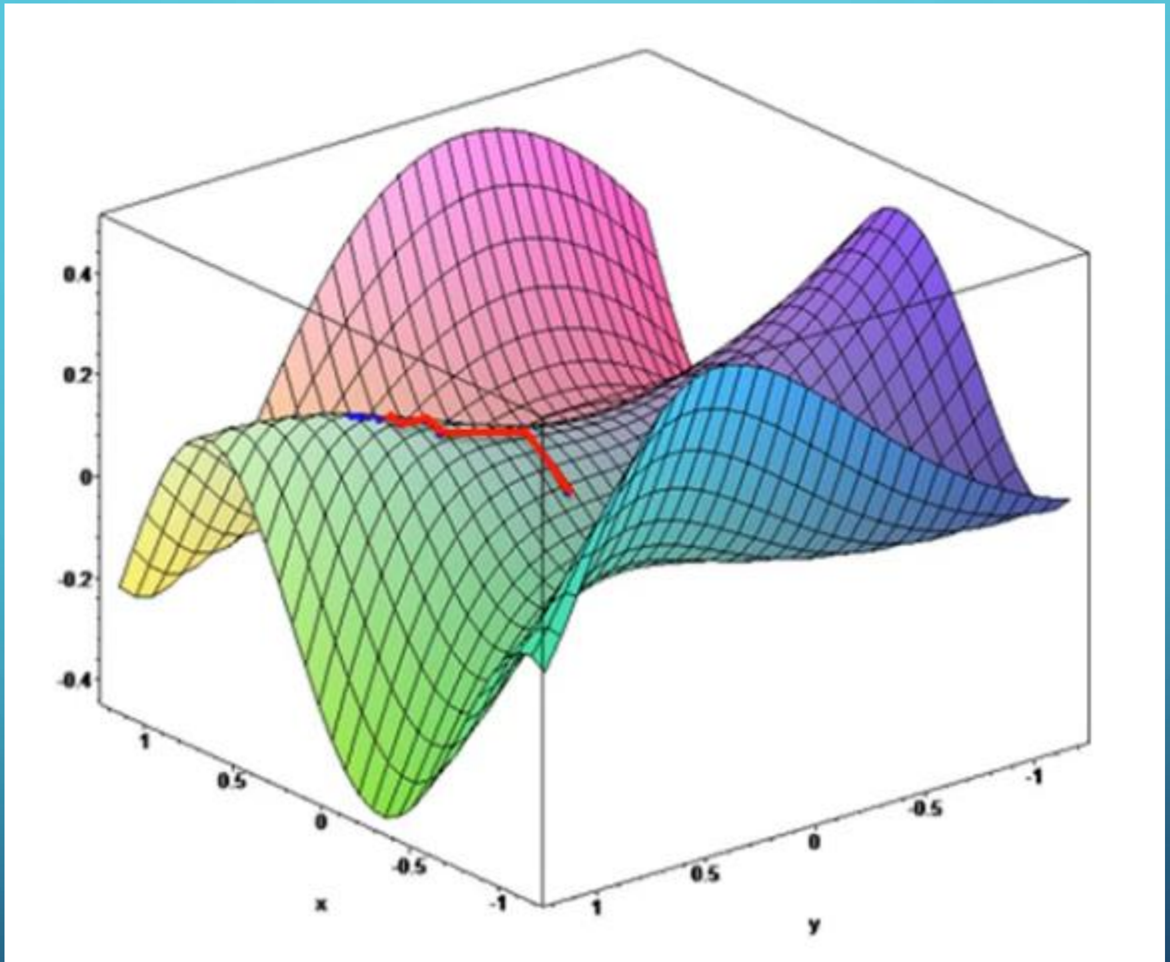



C
ŷ
C = ½(ŷ- y)²


C
C = ½(ŷ- y)²
ŷ


C
ŷ
C = ½(ŷ- y)²


x_0
x_1
x_2
x_3
x_4



0.4
0.2
0
-0.2
-0.4
1
0.5
0
-0.5
-1
x
1
0.5
0
-0.5
-1
y

	Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
	1	12	6	78%	93%
	2	22	6.5	24%	68%
	3	115	4	100%	95%
Upd w's	4	31	9	67%	75%
	5	0	10	58%	51%
	6	5	8	78%	60%
	7	92	6	82%	89%
	8	57	8	91%	97%

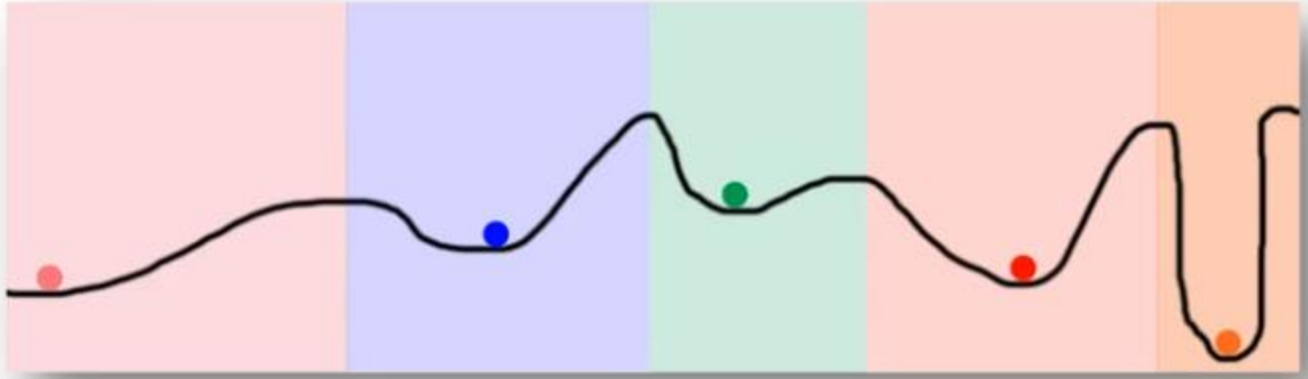
Batch Gradient Descent

	Row ID	Study Hrs	Sleep Hrs	Quiz	Exam
Upd w's	1	12	6	78%	93%
Upd w's	2	22	6.5	24%	68%
Upd w's	3	115	4	100%	95%
Upd w's	4	31	9	67%	75%
Upd w's	5	0	10	58%	51%
Upd w's	6	5	8	78%	60%
Upd w's	7	92	6	82%	89%
Upd w's	8	57	8	91%	97%

Stochastic Gradient Descent

Additional Reading:

A Neural Network in 13 lines of Python (Part 2 - Gradient Descent)



Andrew Trask (2015)

Link:

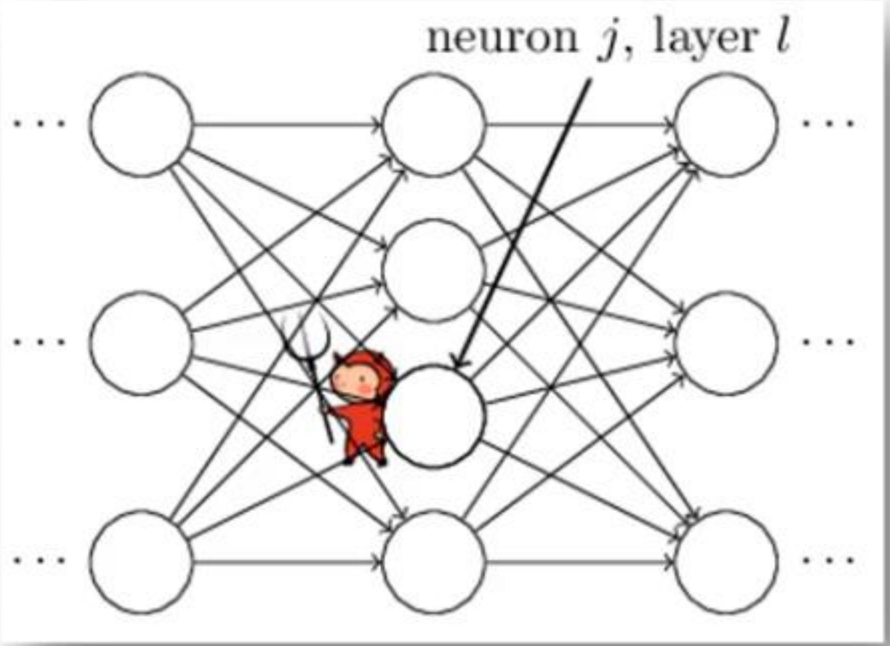
https://iamtrask.github.io/2015/07/27/python-network-part2/

Additional Reading:

Neural Networks and Deep Learning

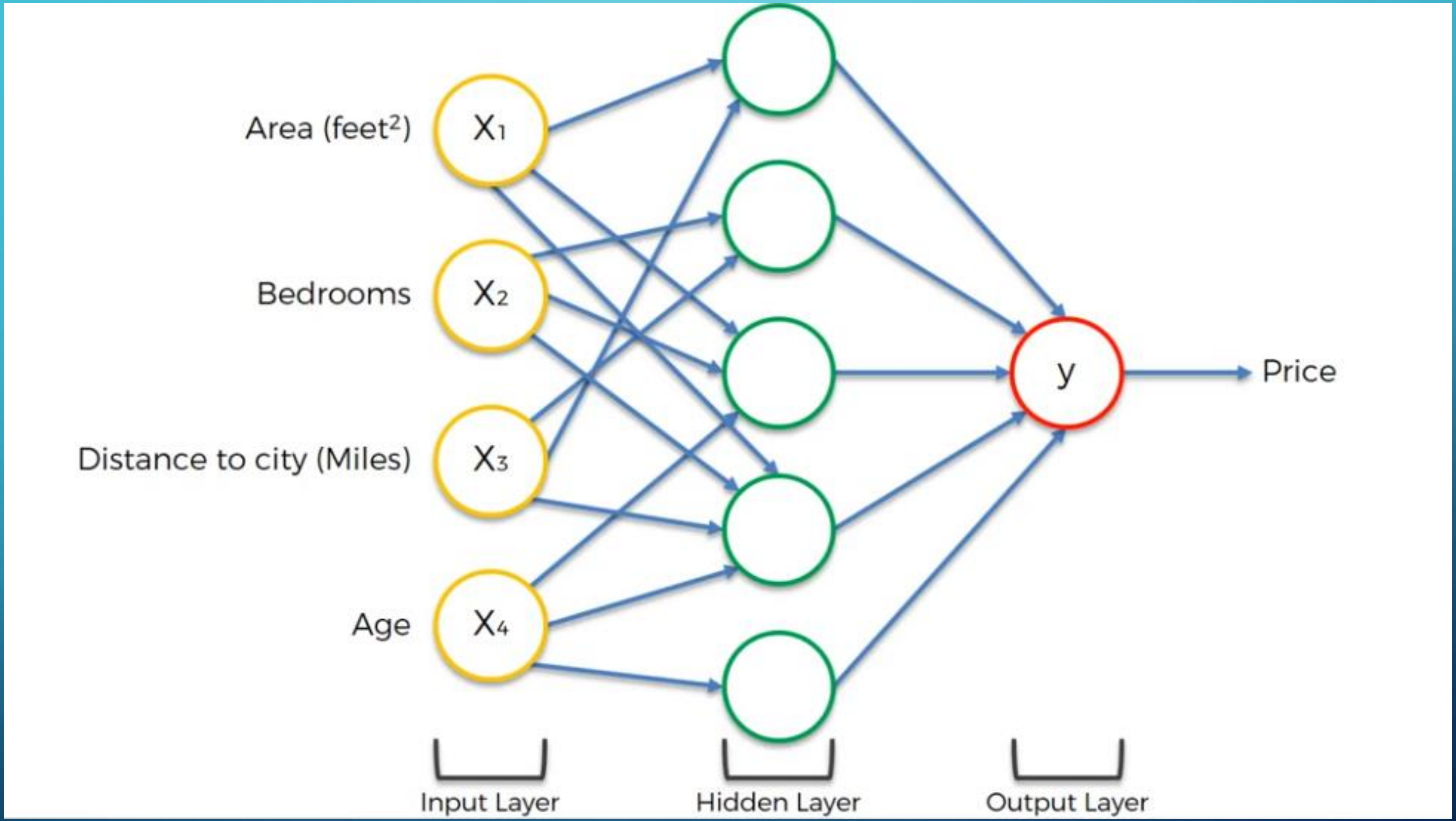
Michael Nielsen (2015)

Link:

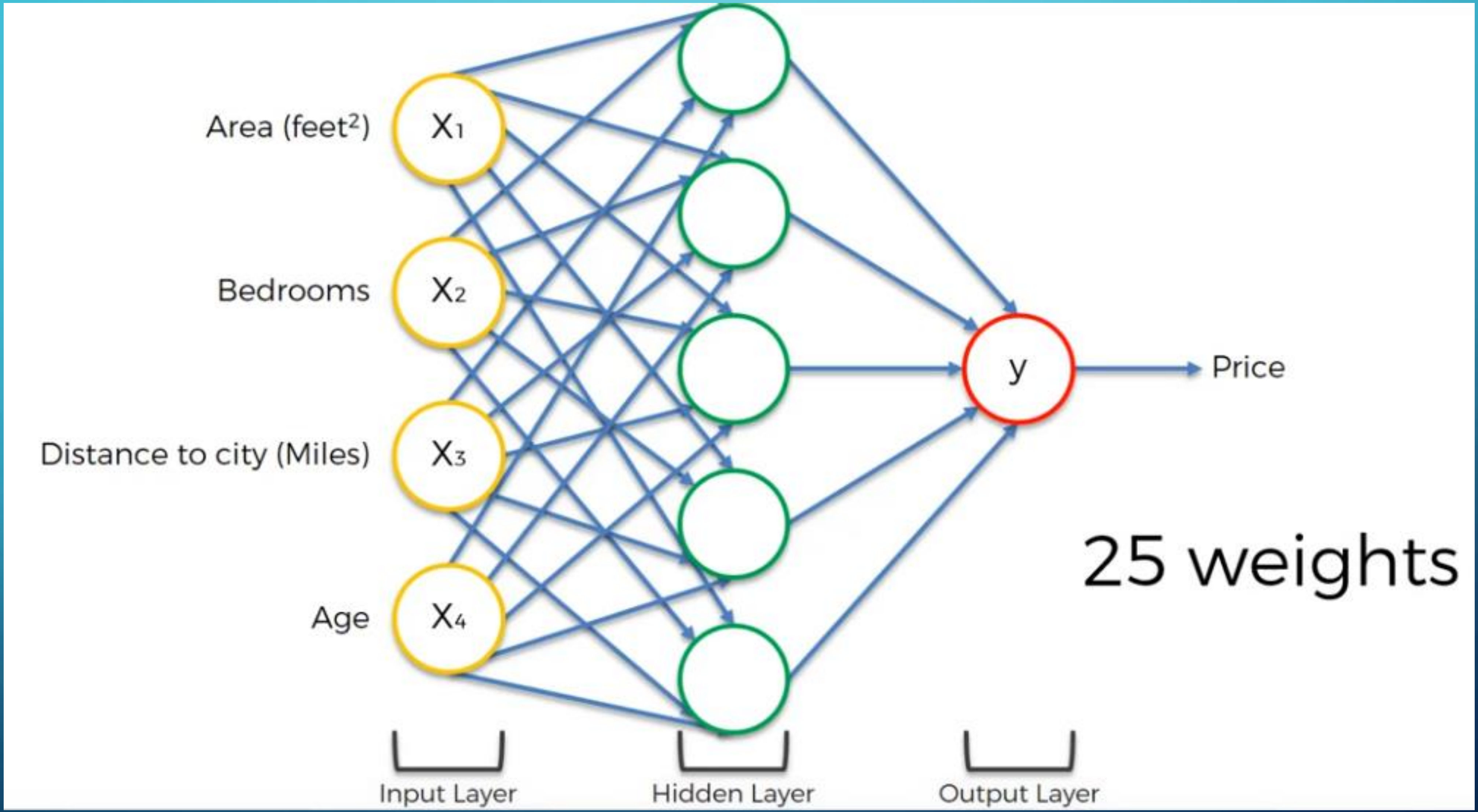


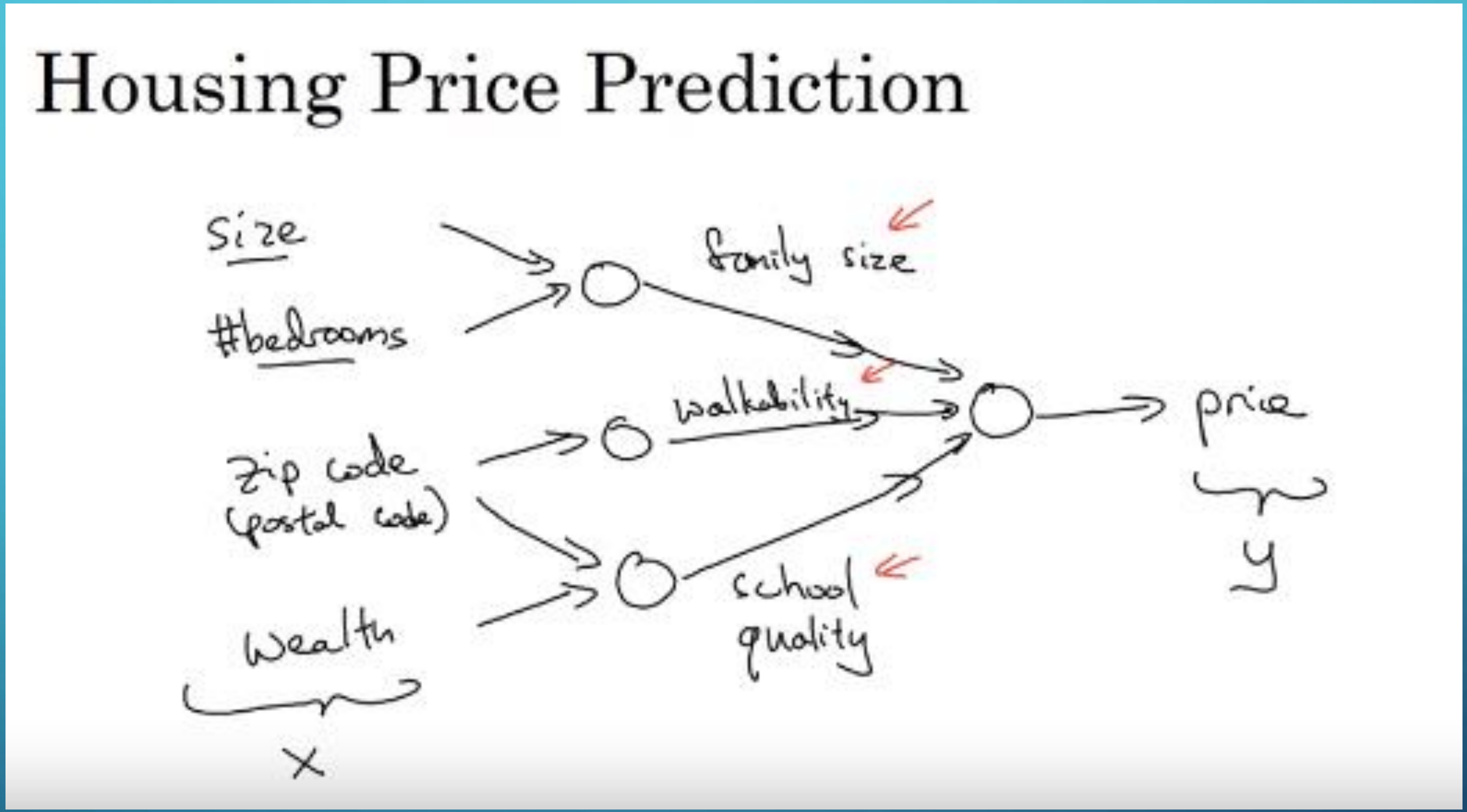
http://neuralnetworksanddeeplearning.com/chap2.html

FORWARD PROPAGATION

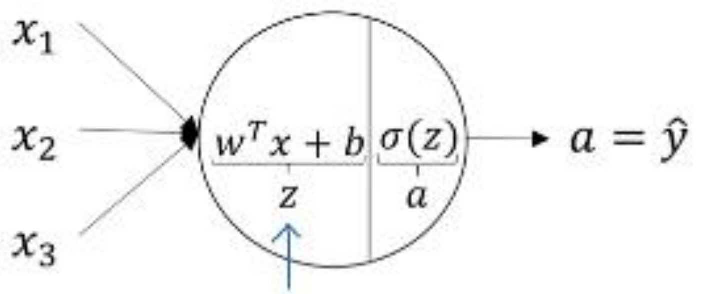


Area (feet²)
Bedrooms
Distance to city (Miles)
Age
X1
X2
X3
X4
y
Price
Input Layer
Hidden Layer
Output Layer


Area (feet²)
Bedrooms
Distance to city (Miles)
Age
X1
X2
X3
X4
y
Price
25 weights
Input Layer
Hidden Layer
Output Layer


Housing Price Prediction
size
#bedrooms
zip code
(postal code)
wealth
x
family size
walkability
school
quality
price
y

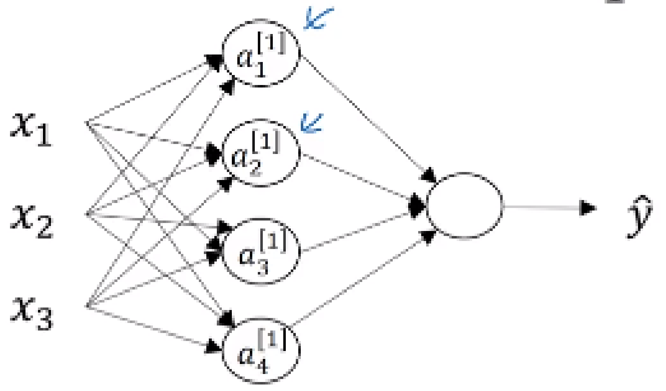
Neural Network Representation



$$z = w^T x + b$$

$$a = \sigma(z)$$

Neural Network Representation



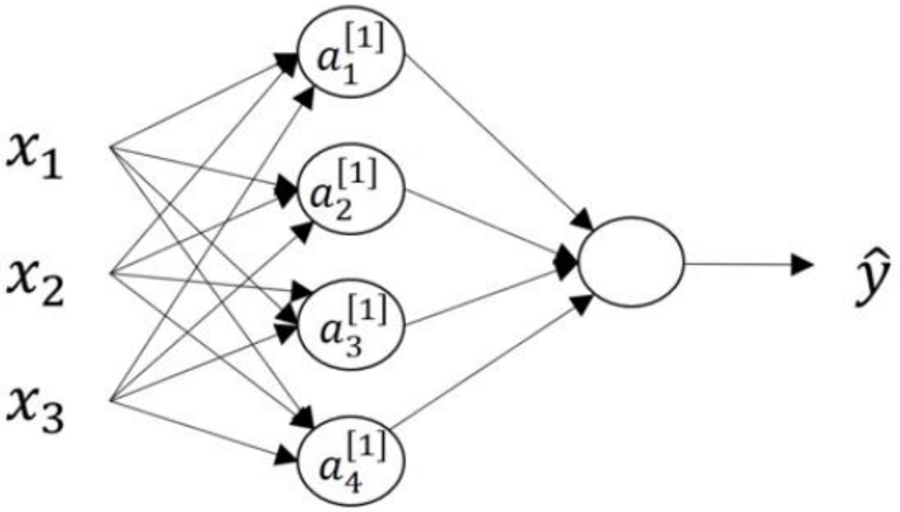
$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]}, \; a_1^{[1]} = \sigma(z_1^{[1]})$$

$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]}, \; a_2^{[1]} = \sigma(z_2^{[1]})$$

$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]}, \; a_3^{[1]} = \sigma(z_3^{[1]})$$

$$z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]}, \; a_4^{[1]} = \sigma(z_4^{[1]})$$

Neural Network Representation learning



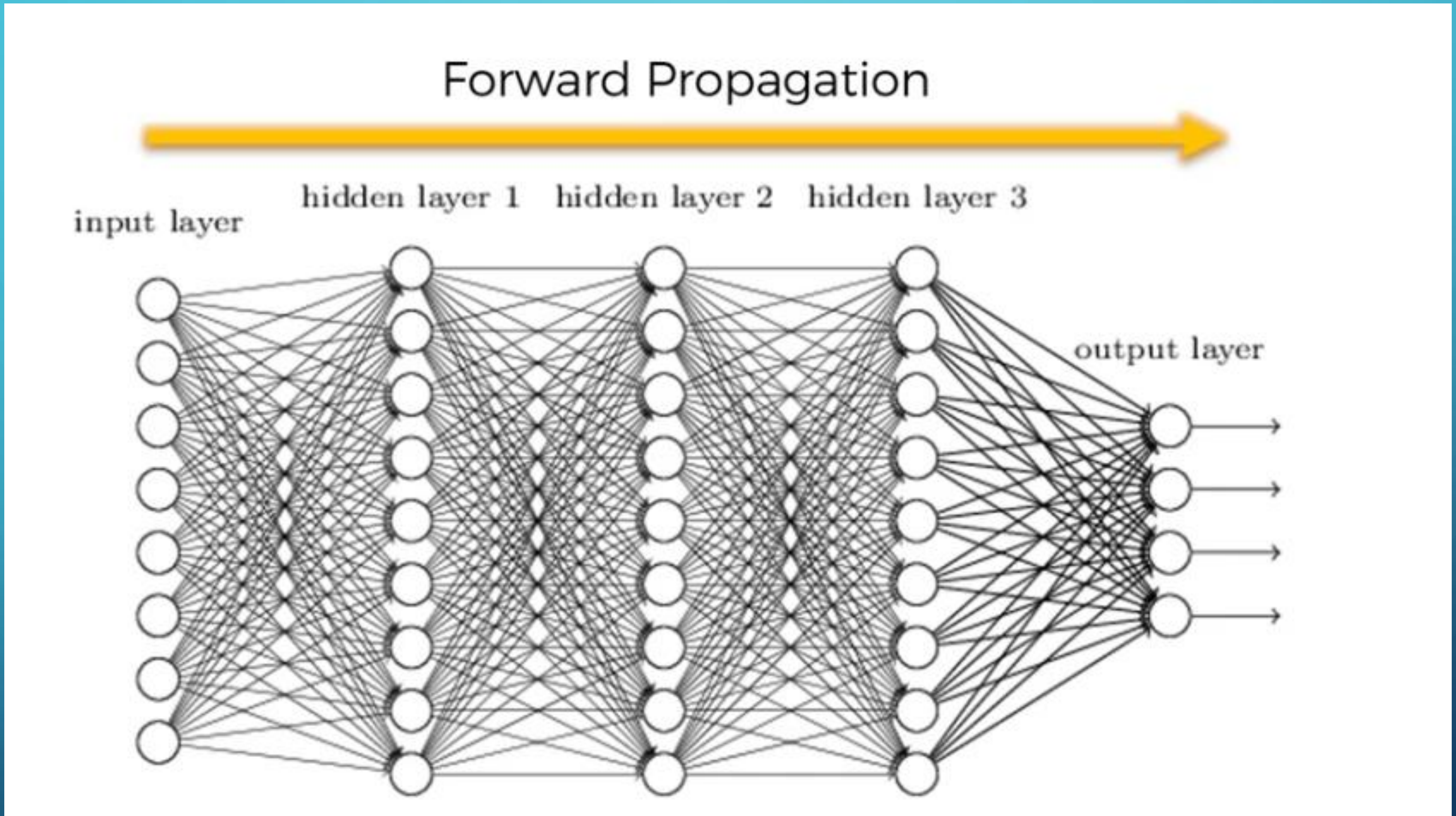
Given input x:

$$z^{[1]} = W^{[1]}x + b^{[1]}$$

$$a^{[1]} = \sigma(z^{[1]})$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

$$a^{[2]} = \sigma(z^{[2]})$$


Forward Propagation
input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

THEN CALCULATE LOSS…

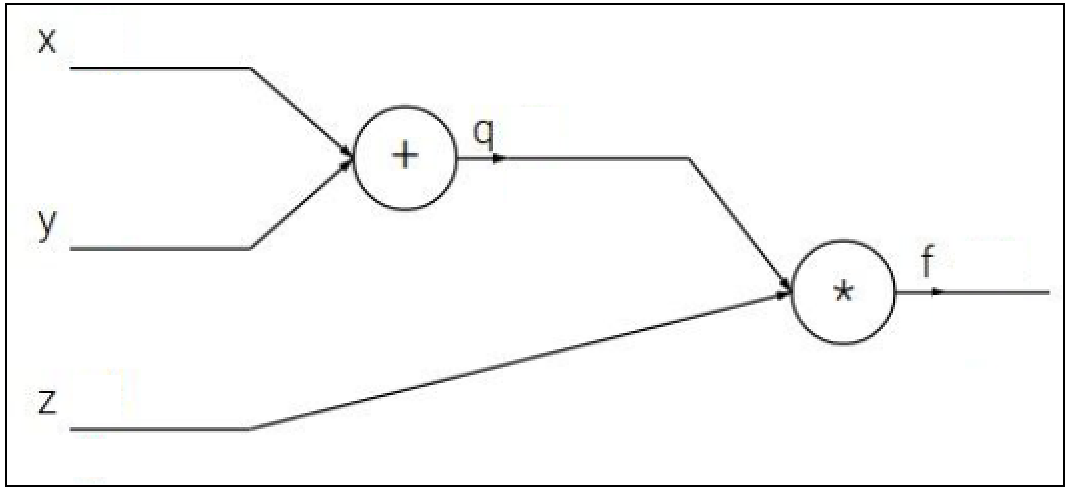
BACKPROPAGATION

- Loss is backpropagated like we did in Linear and Logistic Regression

CHAIN RULE

Backpropagation: a simple example

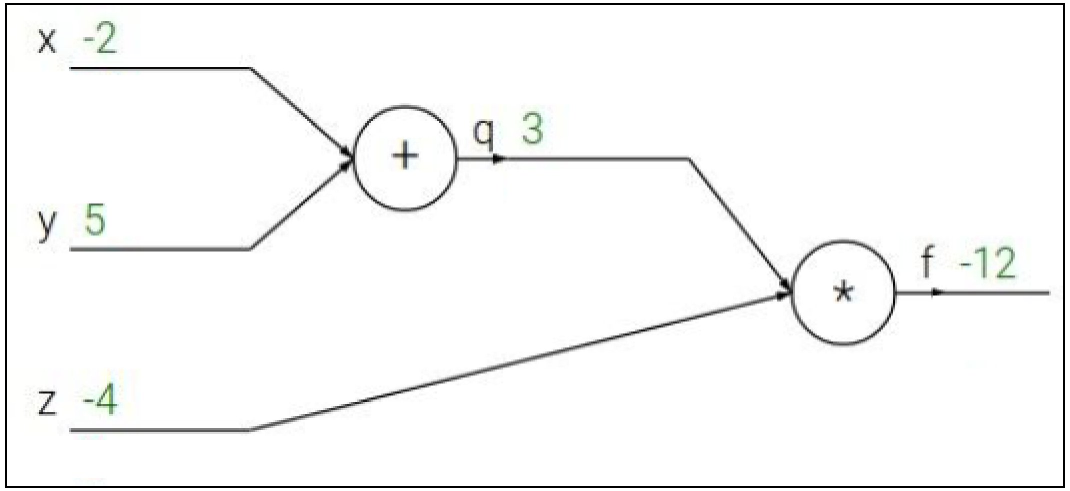
$$f(x, y, z) = (x + y)z$$



Backpropagation: a simple example

$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4



Backpropagation: a simple example

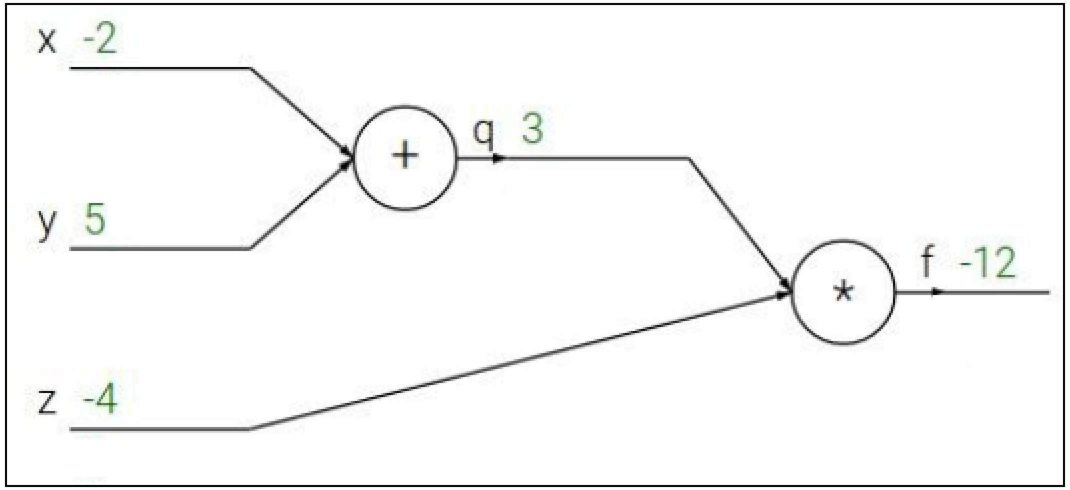
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Backpropagation: a simple example

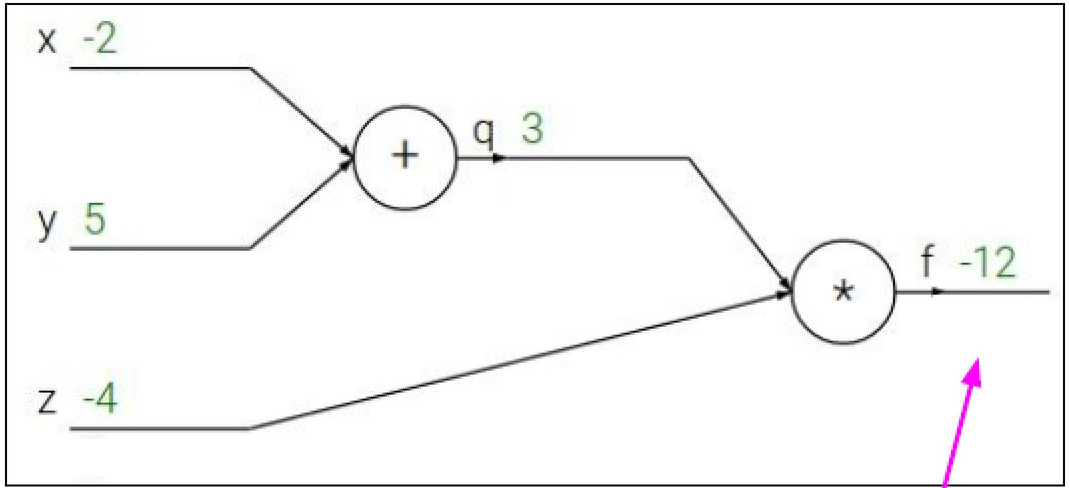
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial f}$$

Backpropagation: a simple example

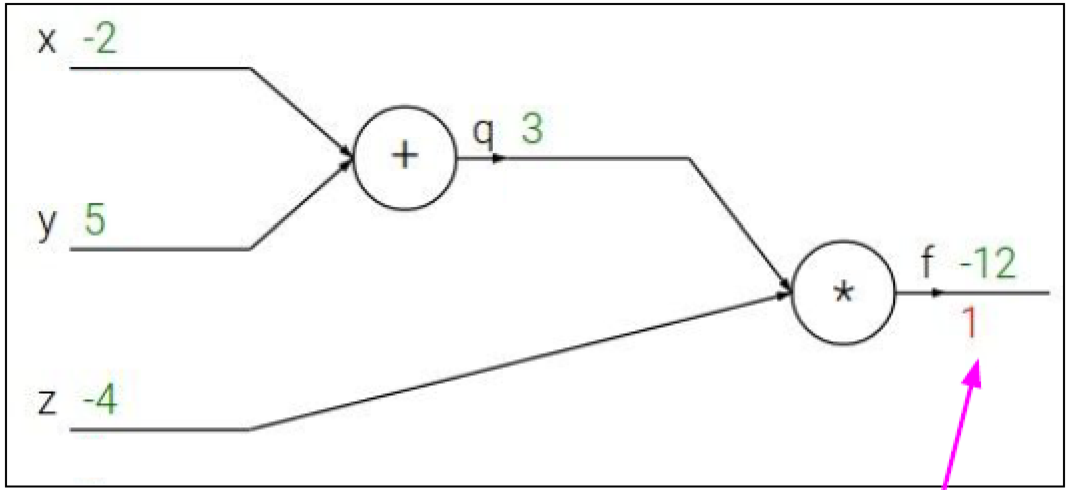
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial f}$$

Backpropagation: a simple example

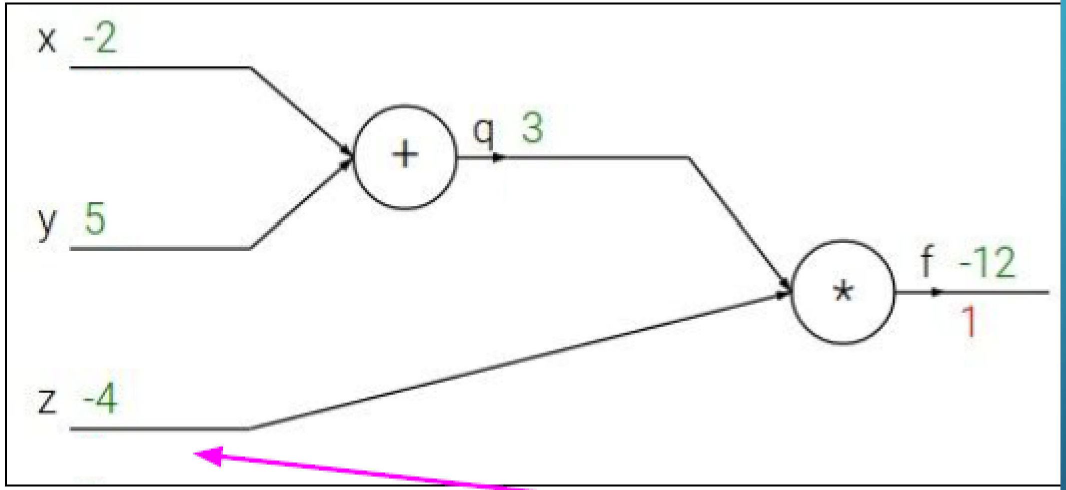
$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

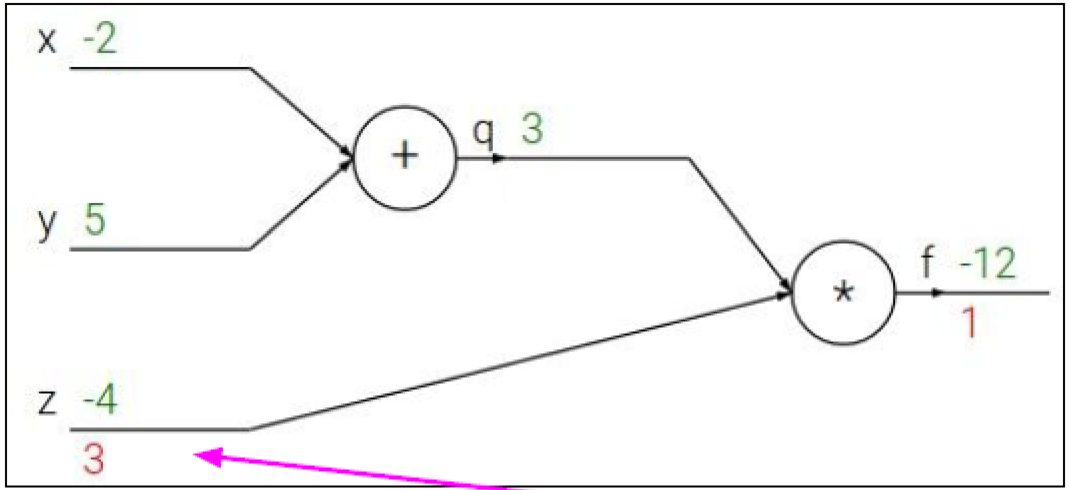


$\frac{\partial f}{\partial z}$

Backpropagation: a simple example

$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4



$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

$\frac{\partial f}{\partial z}$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

Backpropagation: a simple example

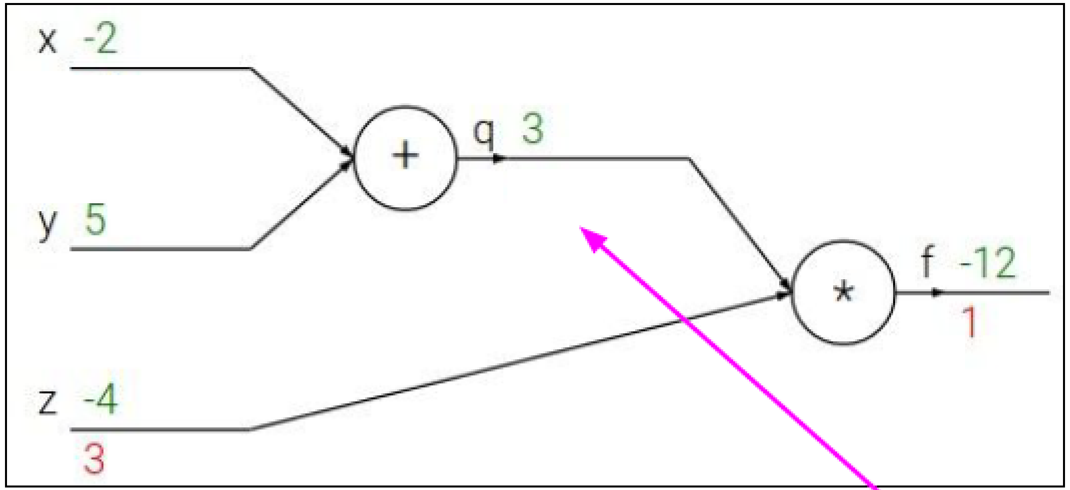
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\dfrac{\partial f}{\partial x}, \dfrac{\partial f}{\partial y}, \dfrac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial q}$$

Backpropagation: a simple example

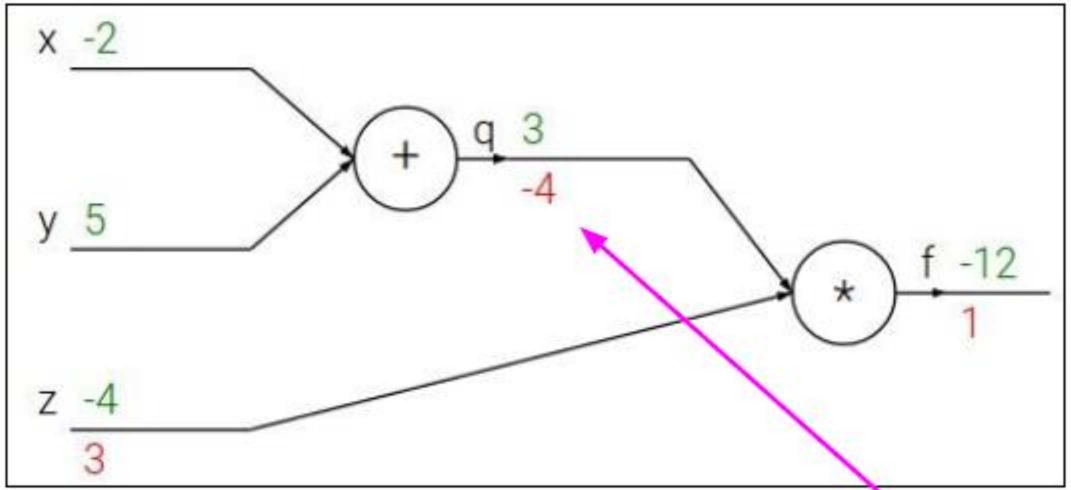
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\boxed{\frac{\partial f}{\partial q}}$$

Backpropagation: a simple example

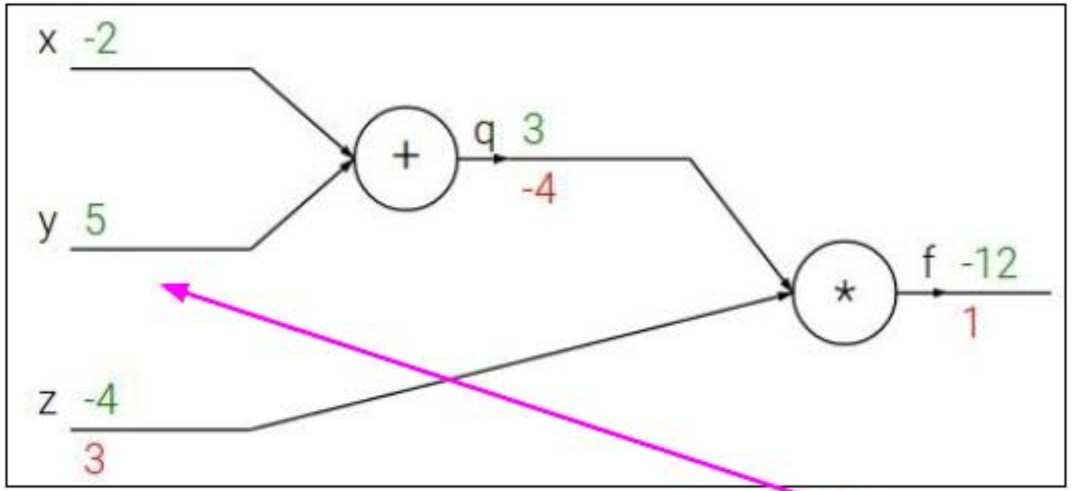
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial y}$$

Chain rule:

$$\boxed{\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y}}$$

Upstream gradient

Local gradient

Backpropagation: a simple example

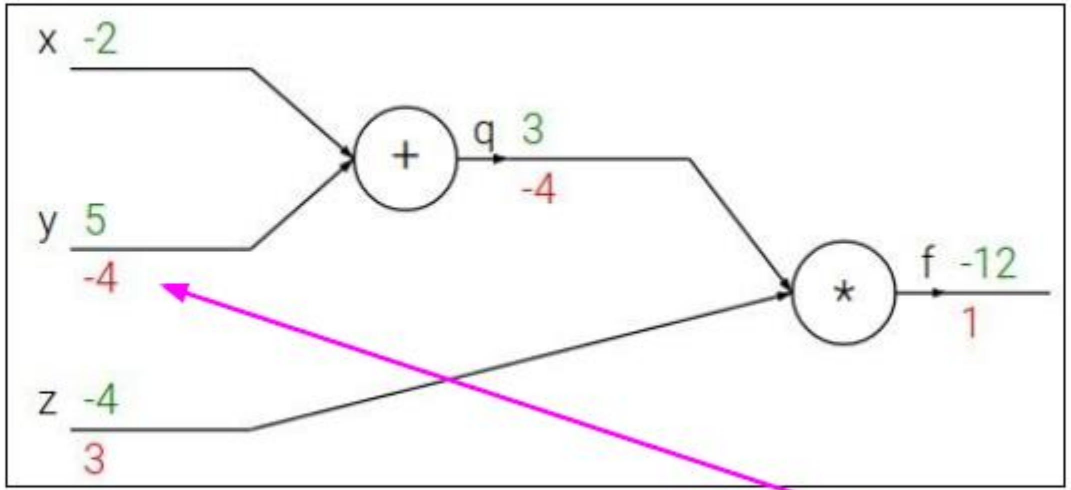
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$\boxed{q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1}$$

$$\boxed{f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q}$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\boxed{\frac{\partial f}{\partial y}}$$

Chain rule:

$$\boxed{\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y}}$$

Upstream gradient

Local gradient

SIMILARLY

Backpropagation: a simple example

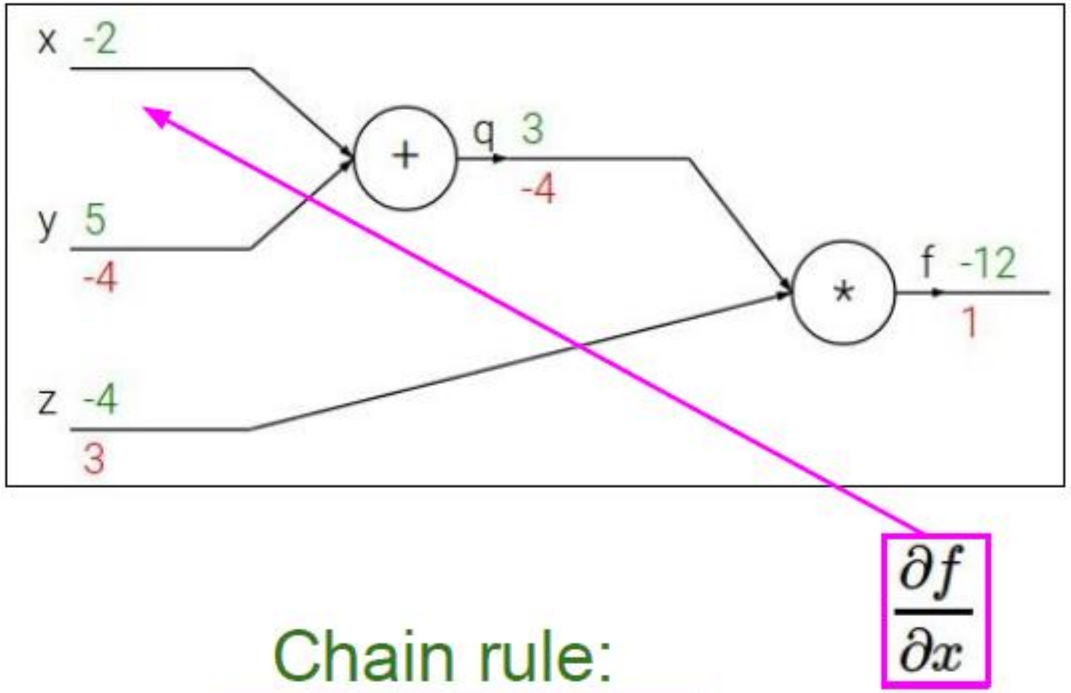
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Chain rule:

$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}}$$

Upstream gradient

Local gradient

Backpropagation: a simple example

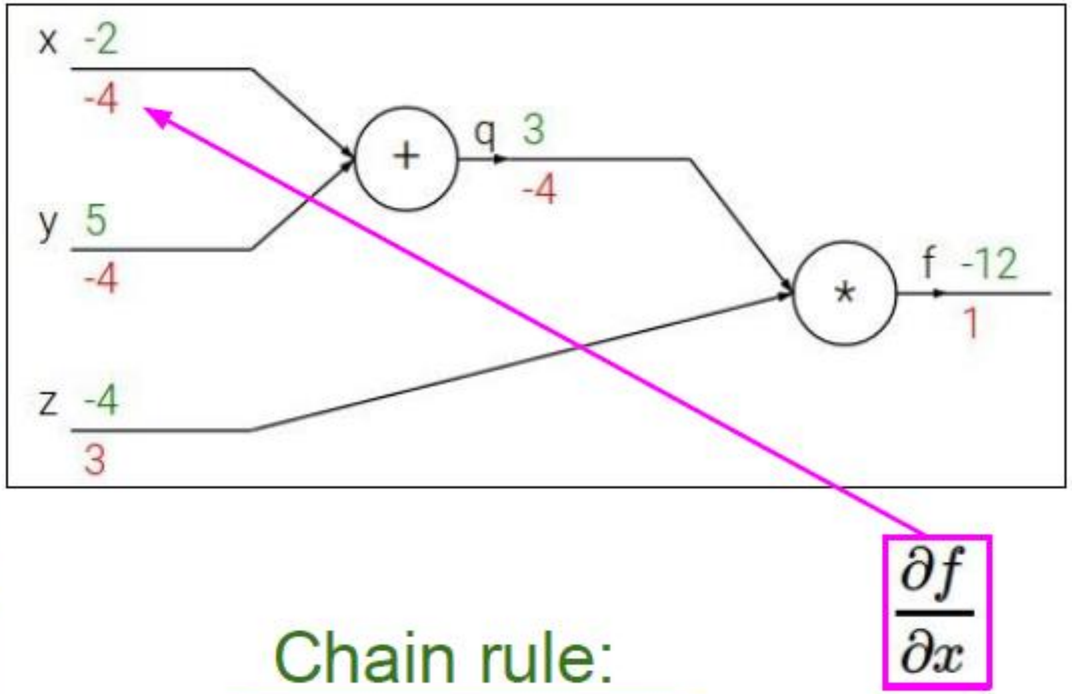
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

$$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

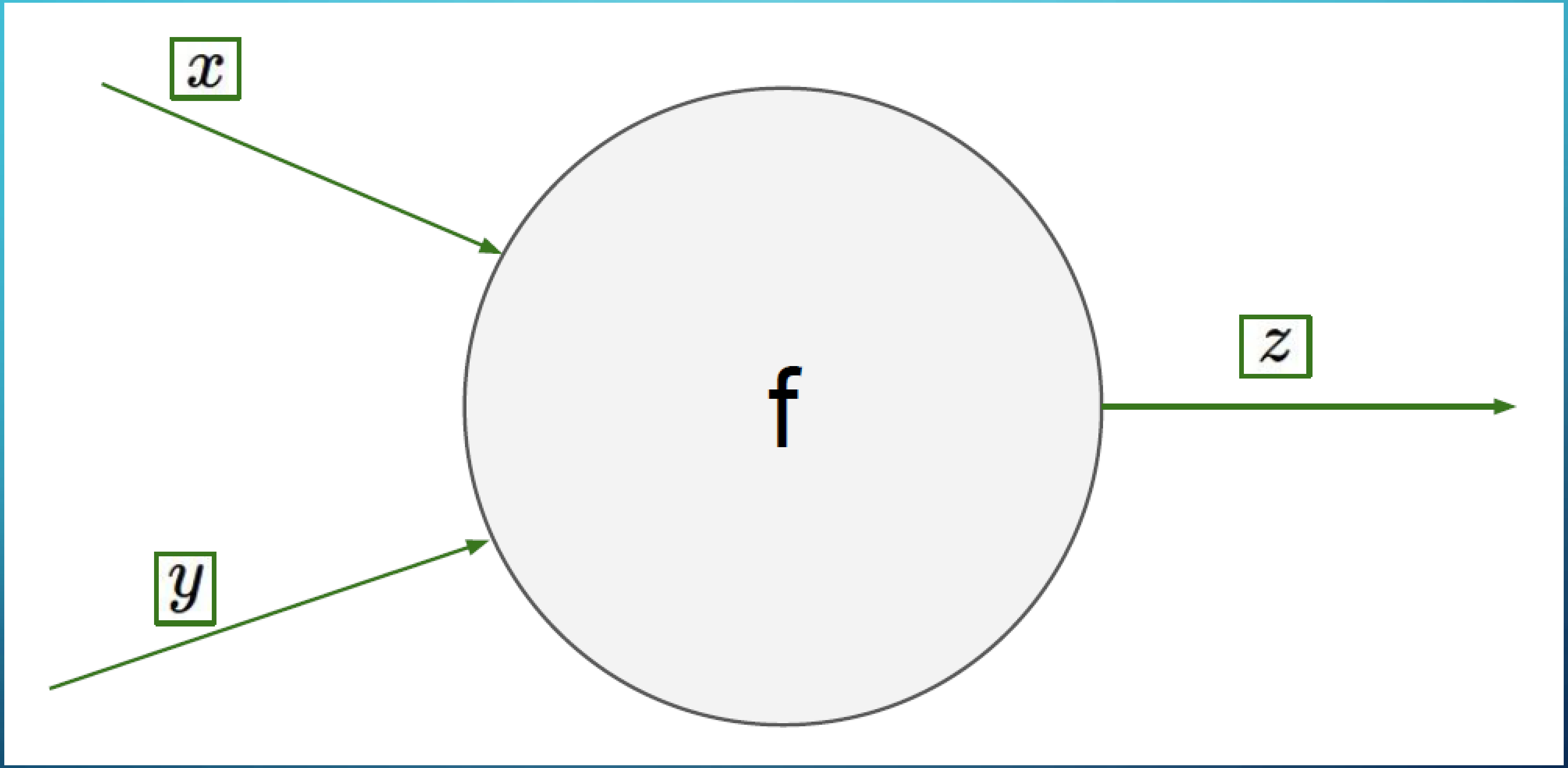


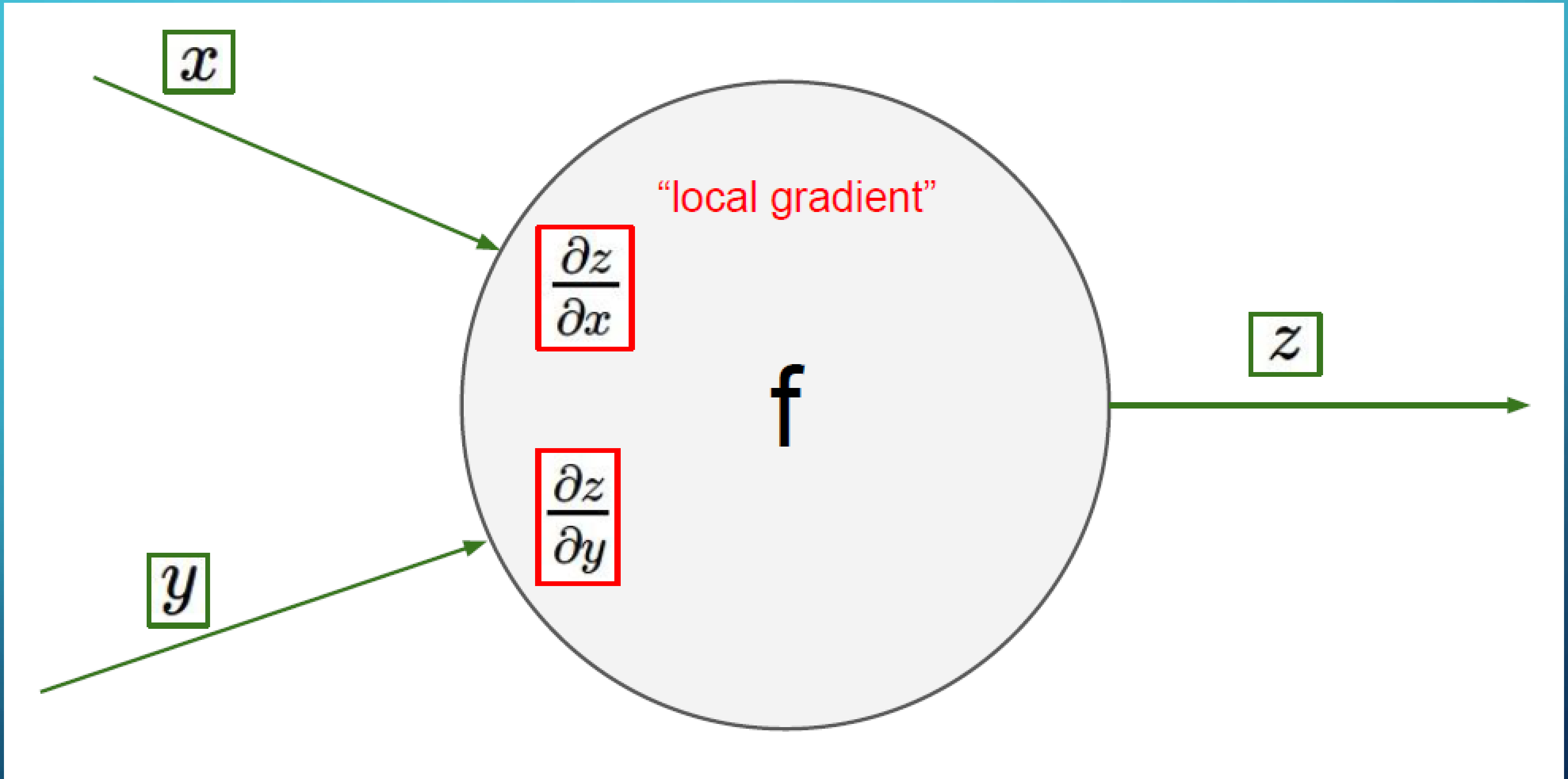
Chain rule:

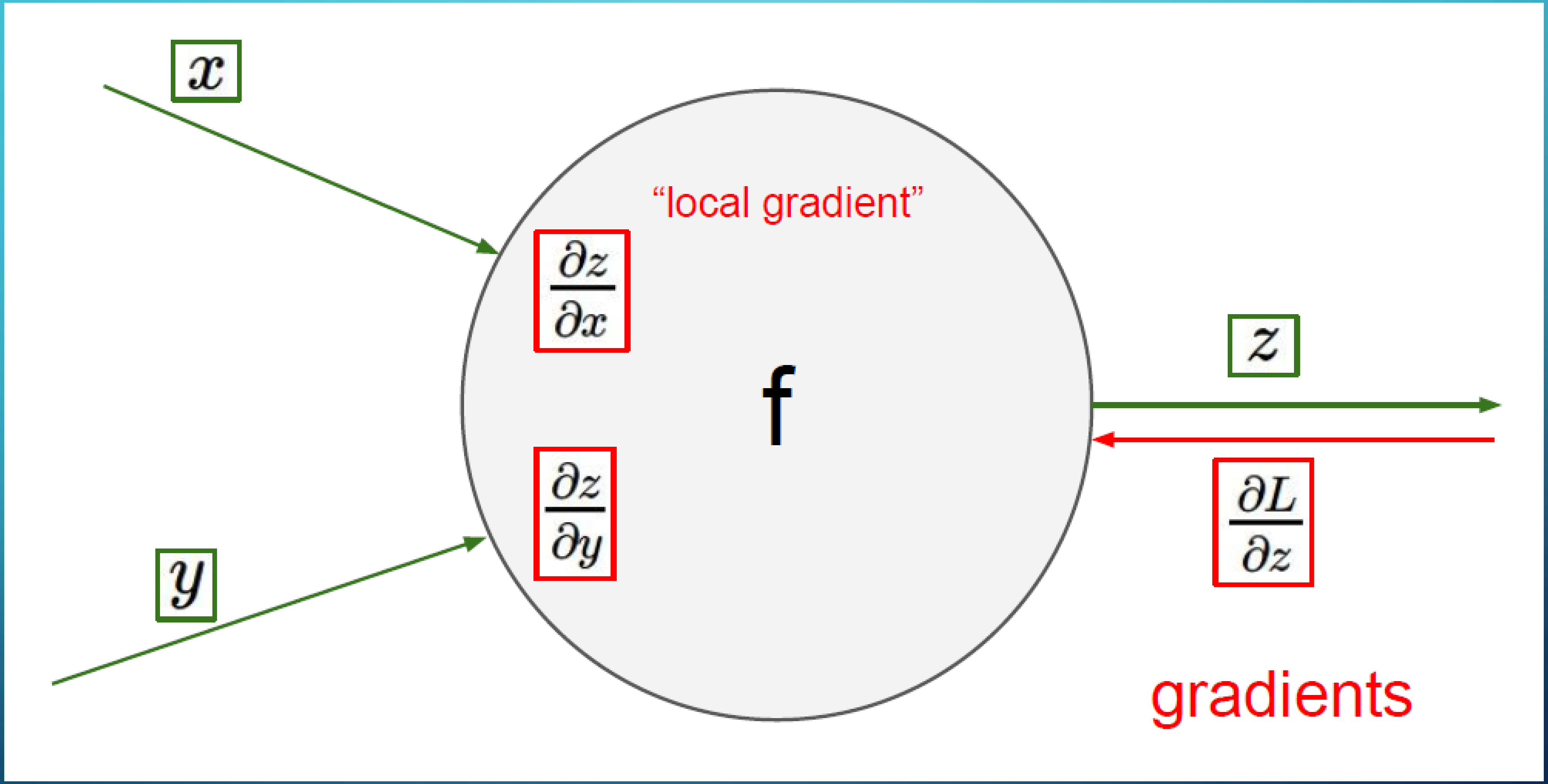
$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}}$$

Upstream gradient

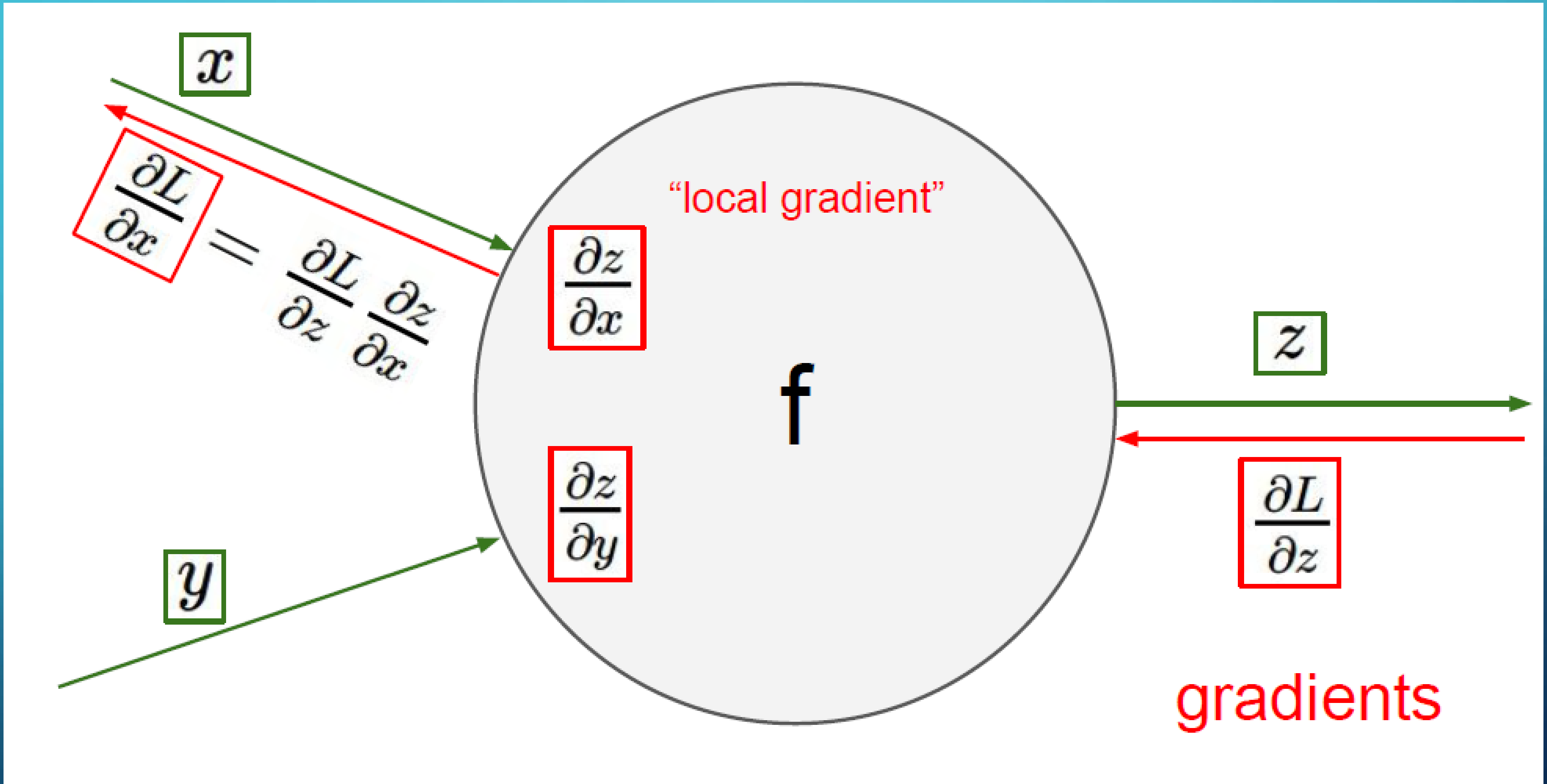
Local gradient


x
y
f
z

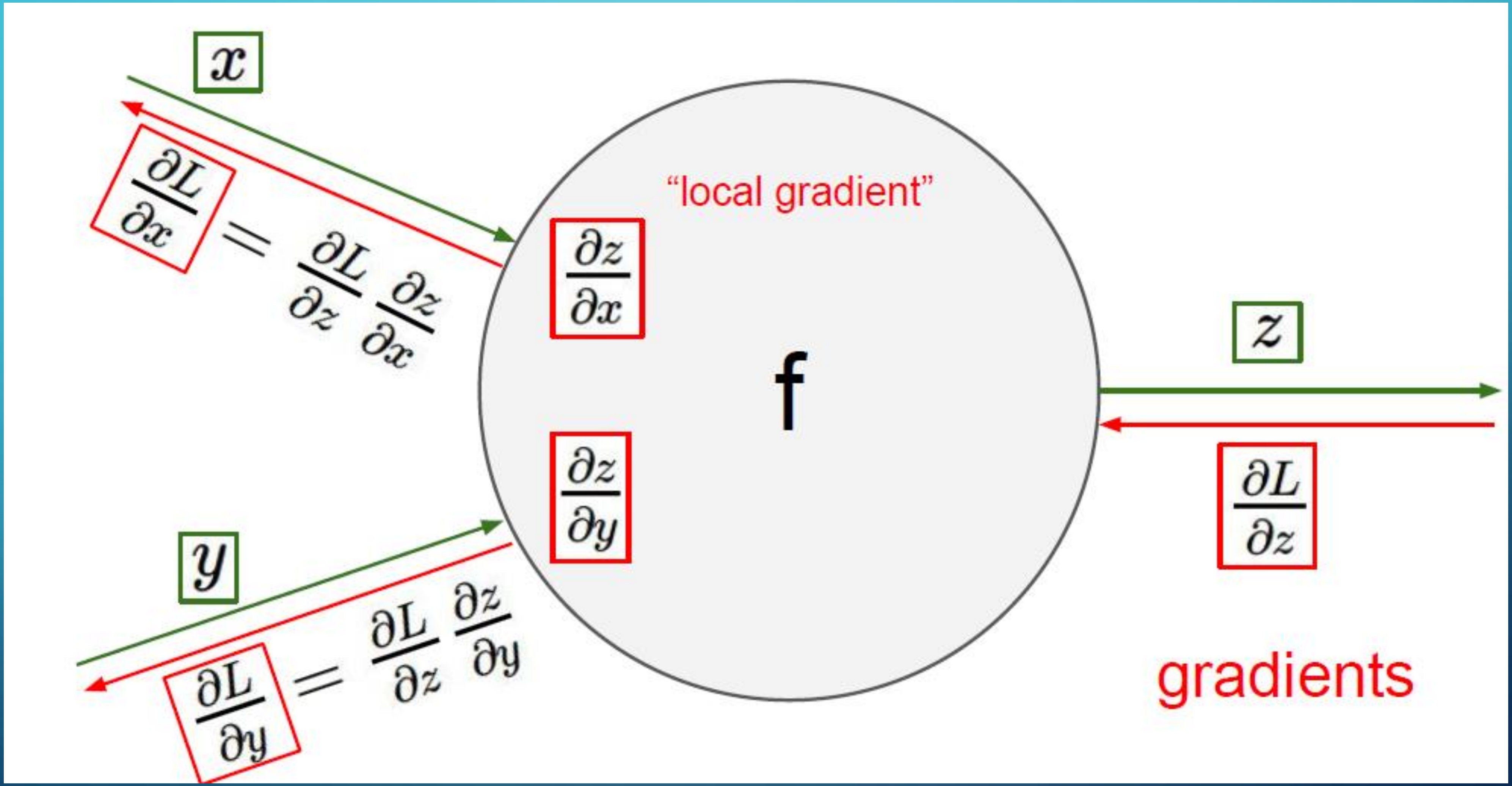

x
"local gradient"
$\frac{\partial z}{\partial x}$
f
z
$\frac{\partial z}{\partial y}$
y



x
y
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
f
z
$\frac{\partial L}{\partial z}$
gradients



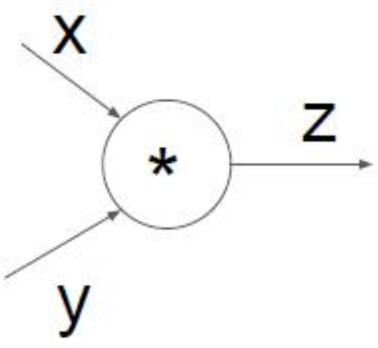
x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
"local gradient"
$\frac{\partial z}{\partial x}$
f
$\frac{\partial z}{\partial y}$
y
z
$\frac{\partial L}{\partial z}$
gradients



x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
"local gradient"
$\frac{\partial z}{\partial x}$
f
$\frac{\partial z}{\partial y}$
z
$\frac{\partial L}{\partial z}$
y
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
gradients

THIS STEP IS PERFORMED FOR ALL THE WEIGHTS

Modularized implementation: forward / backward API



(x,y,z are scalars)

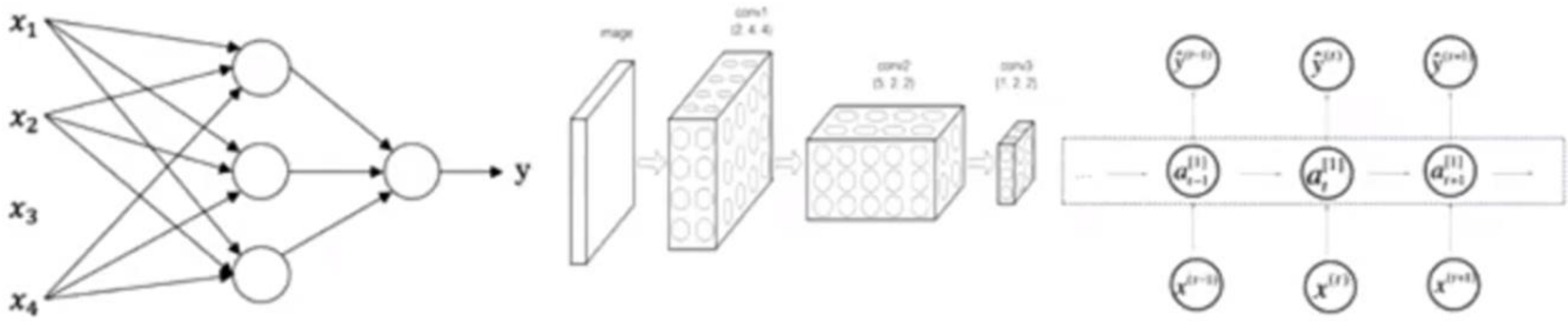
```
class MultiplyGate(object):
    def forward(x,y):
        z = x*y
        self.x = x # must keep these around!
        self.y = y
        return z
    def backward(dz):
        dx = self.y * dz # [dz/dx * dL/dz]
        dy = self.x * dz # [dz/dy * dL/dz]
        return [dx, dy]
```

Local gradient

Upstream gradient variable

TYPES OF NEURAL NETWORKS

Neural Network examples

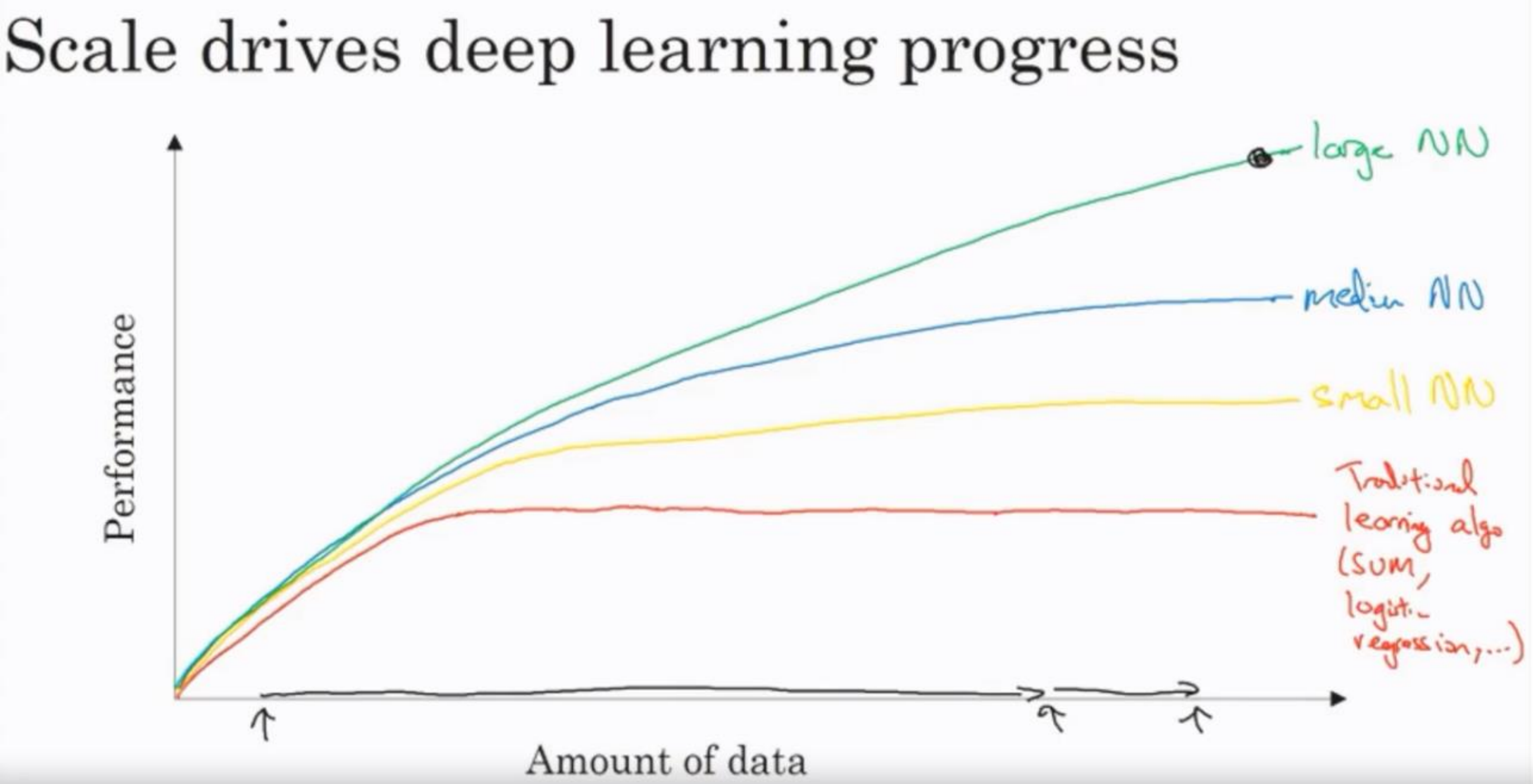


Standard NN Convolutional NN Recurrent NN

APPLICATIONS

Input(x)	Output (y)	Application	
Home features	Price	Real Estate	Standard NN
Ad, user info	Click on ad? (0/1)	Online Advertising	
Image	Object (1,…,1000)	Photo tagging	CNN
Audio	Text transcript	Speech recognition	RNN
English	Chinese	Machine translation	
Image, Radar info	Position of other cars	Autonomous driving	Custom/ Hybrid

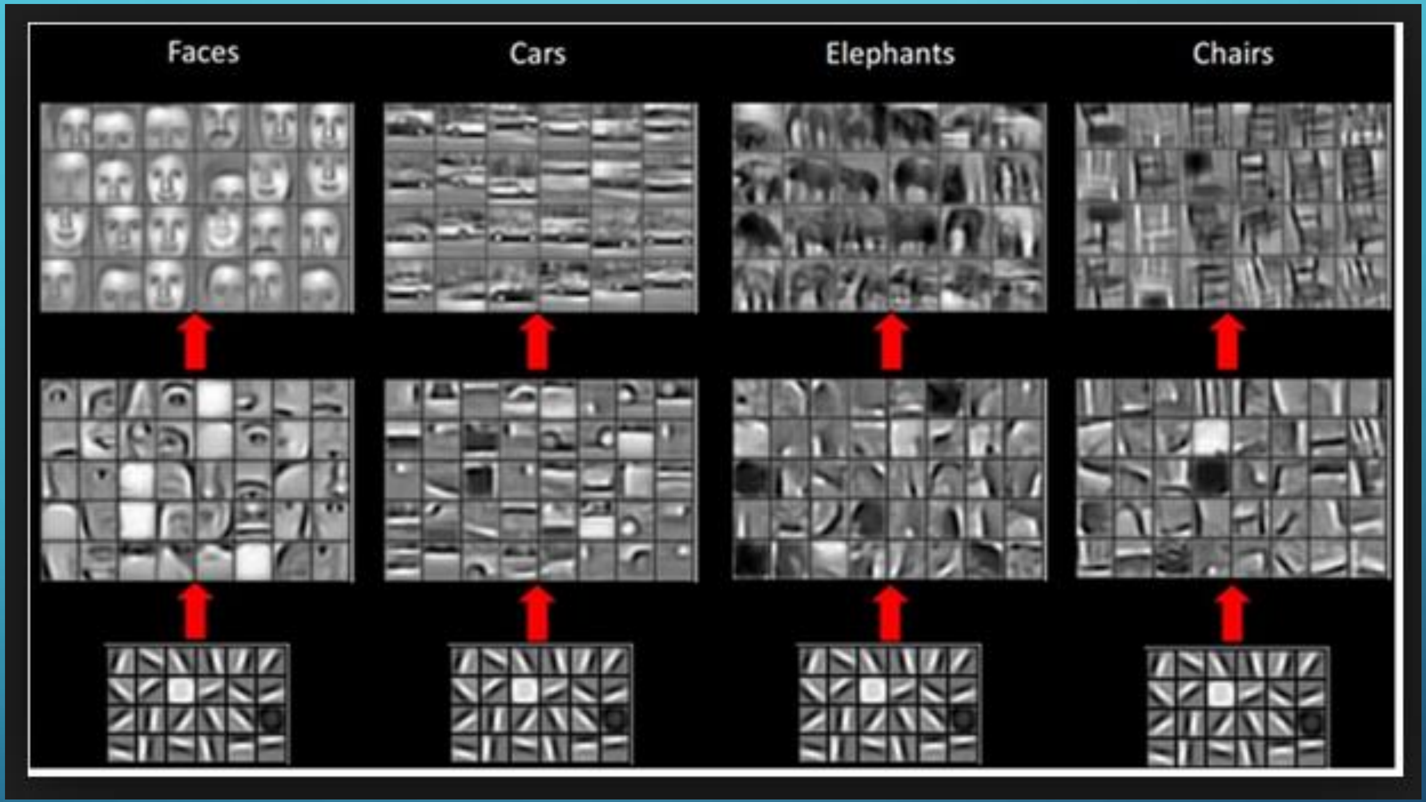
WHY DEEP LEARNING IS TAKING OFF?



Scale drives deep learning progress

- Data
- Computation
- Algorithms

TIME FOR GOOGLE PLAYGROUND


Faces
Cars
Elephants
Chairs

PUT ON YOUR THINKING HAT!!

Time to get your hands dirty!!