SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of AI &DS

Course Name – 19AD602 DEEP LEARNING

III Year /VI Semester

Unit 1-INTRODUCTION
Topic: Training a network

TRAINING A NETWORK

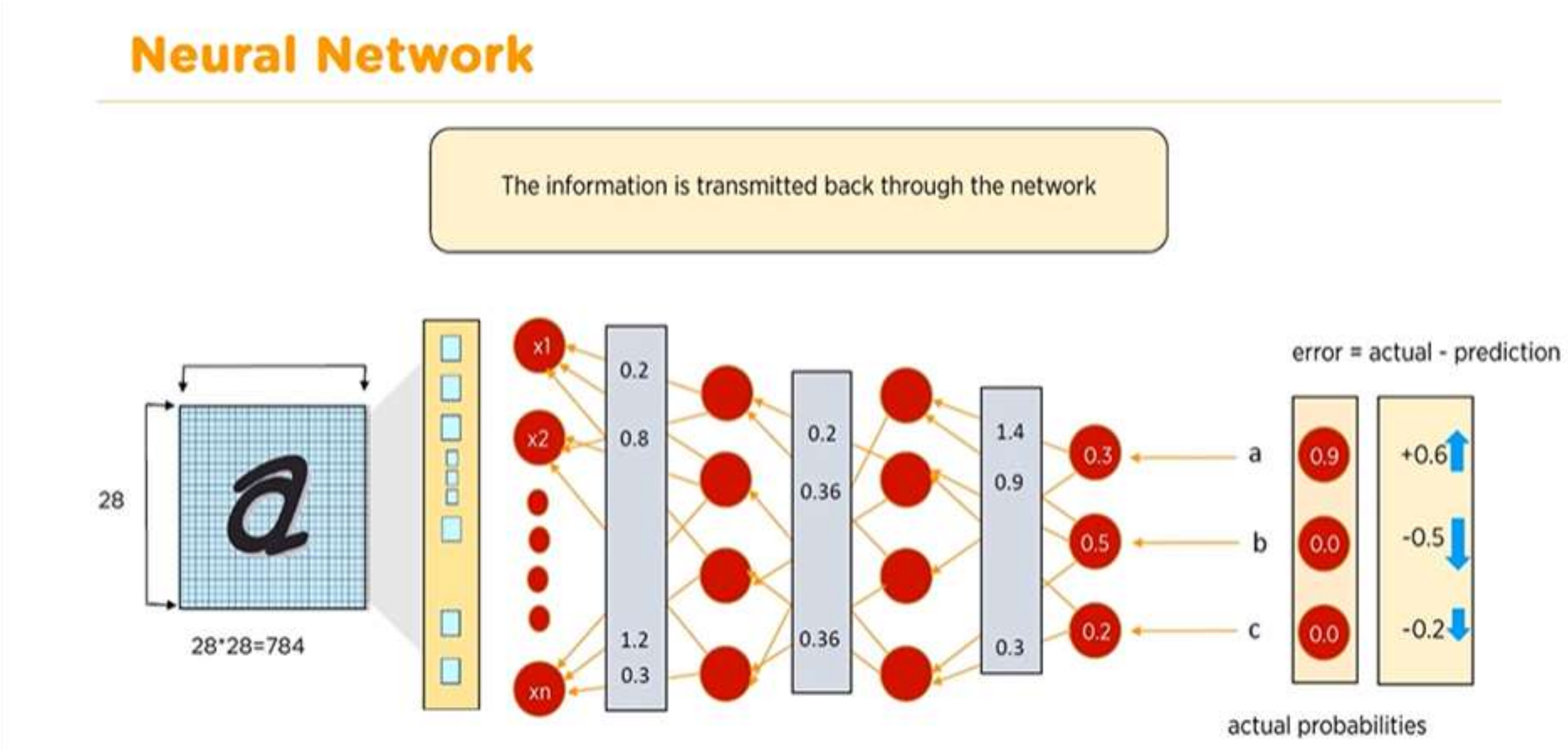
Case Study

Task: Train a network to predict house prices based on size and location.

Approach: Use mean squared error (MSE) as the loss function, update weights using backpropagation, and optimize using stochastic gradient descent.

Outcome: The network minimizes the error and predicts prices with 90% accuracy on a test dataset.

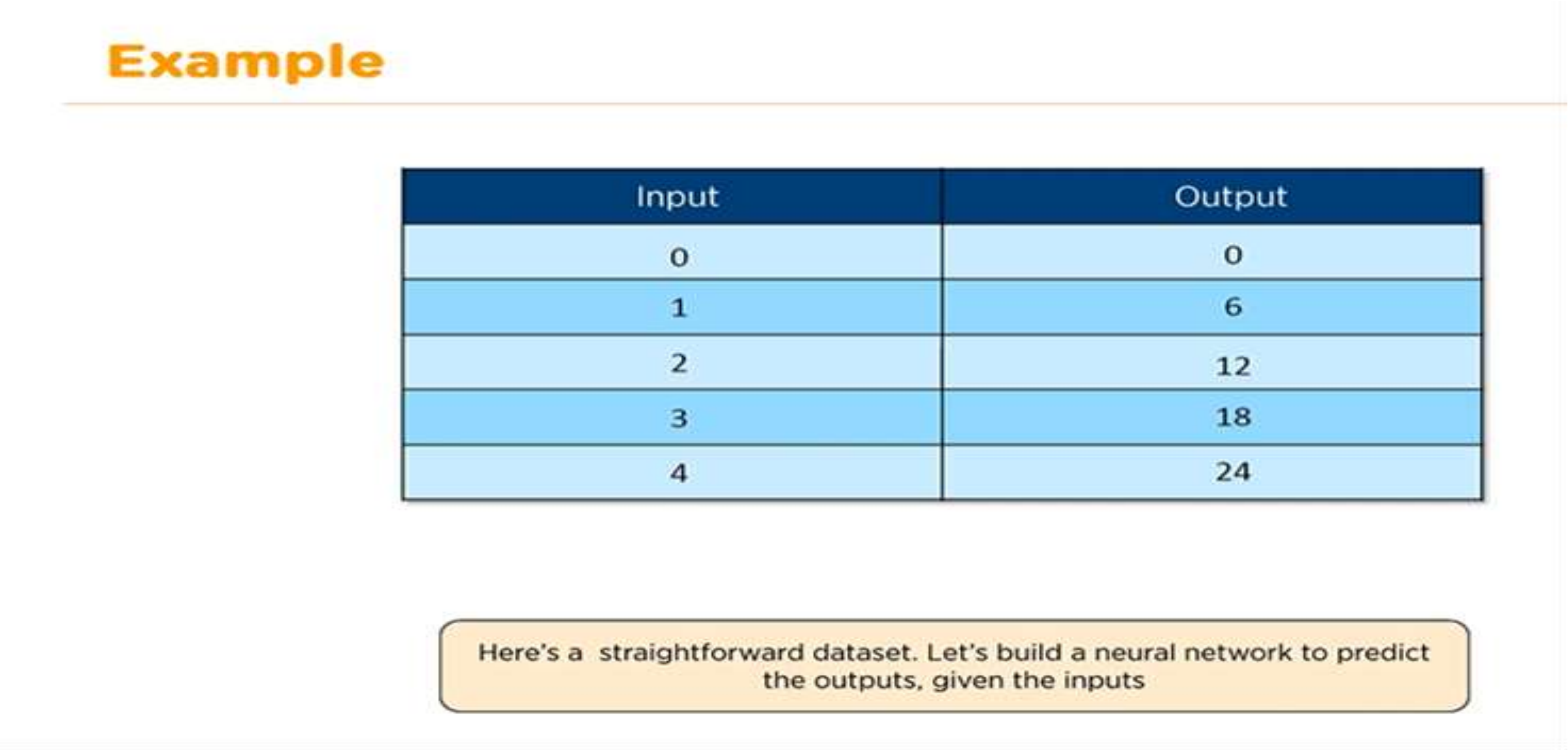
TRAINING A NETWORK



Neural Network

The information is transmitted back through the network

TRAINING A NETWORK

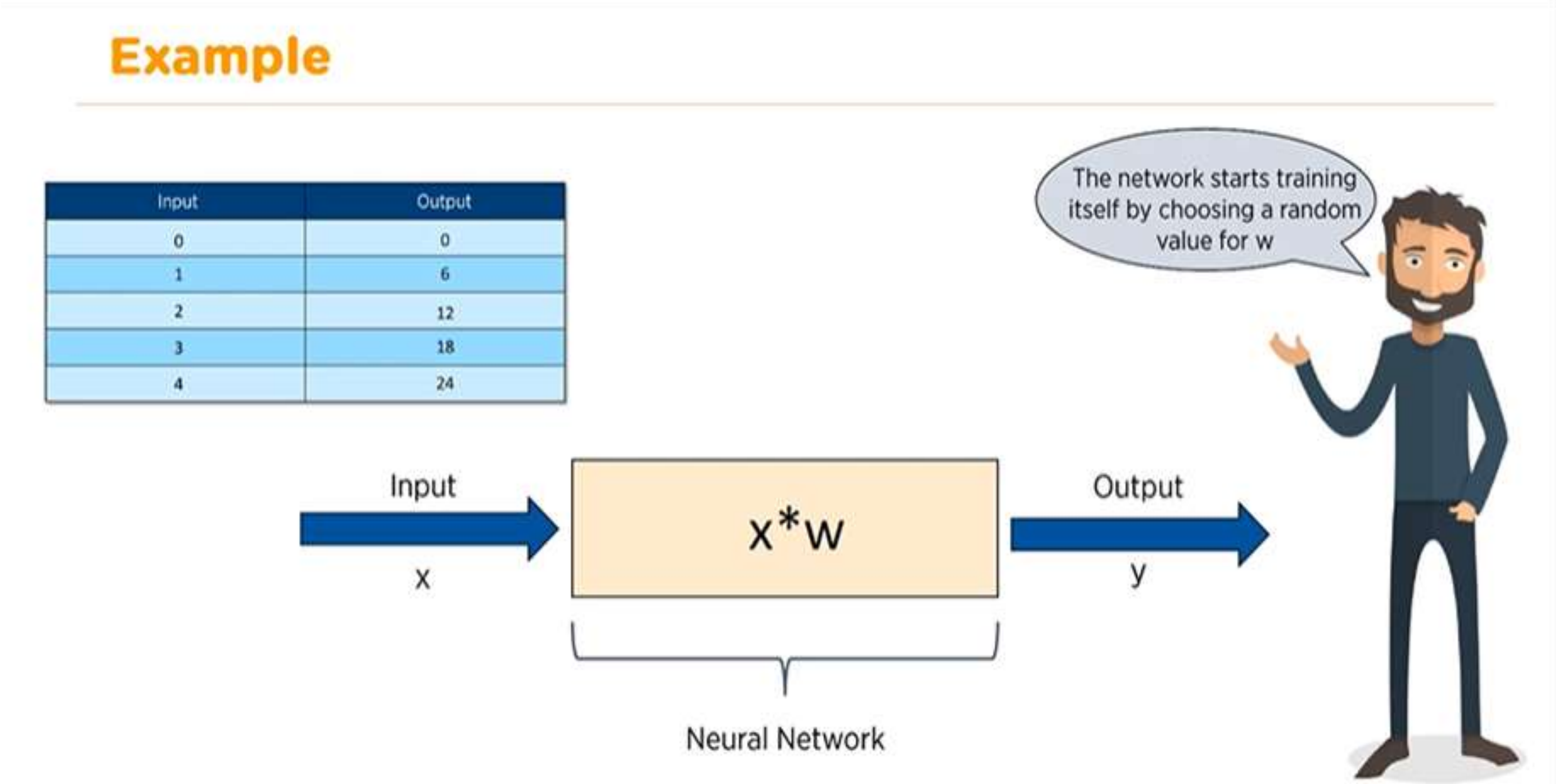


Example

Input	Output
0	0
1	6
2	12
3	18
4	24

Here's a straightforward dataset. Let's build a neural network to predict the outputs, given the inputs

TRAINING A NETWORK

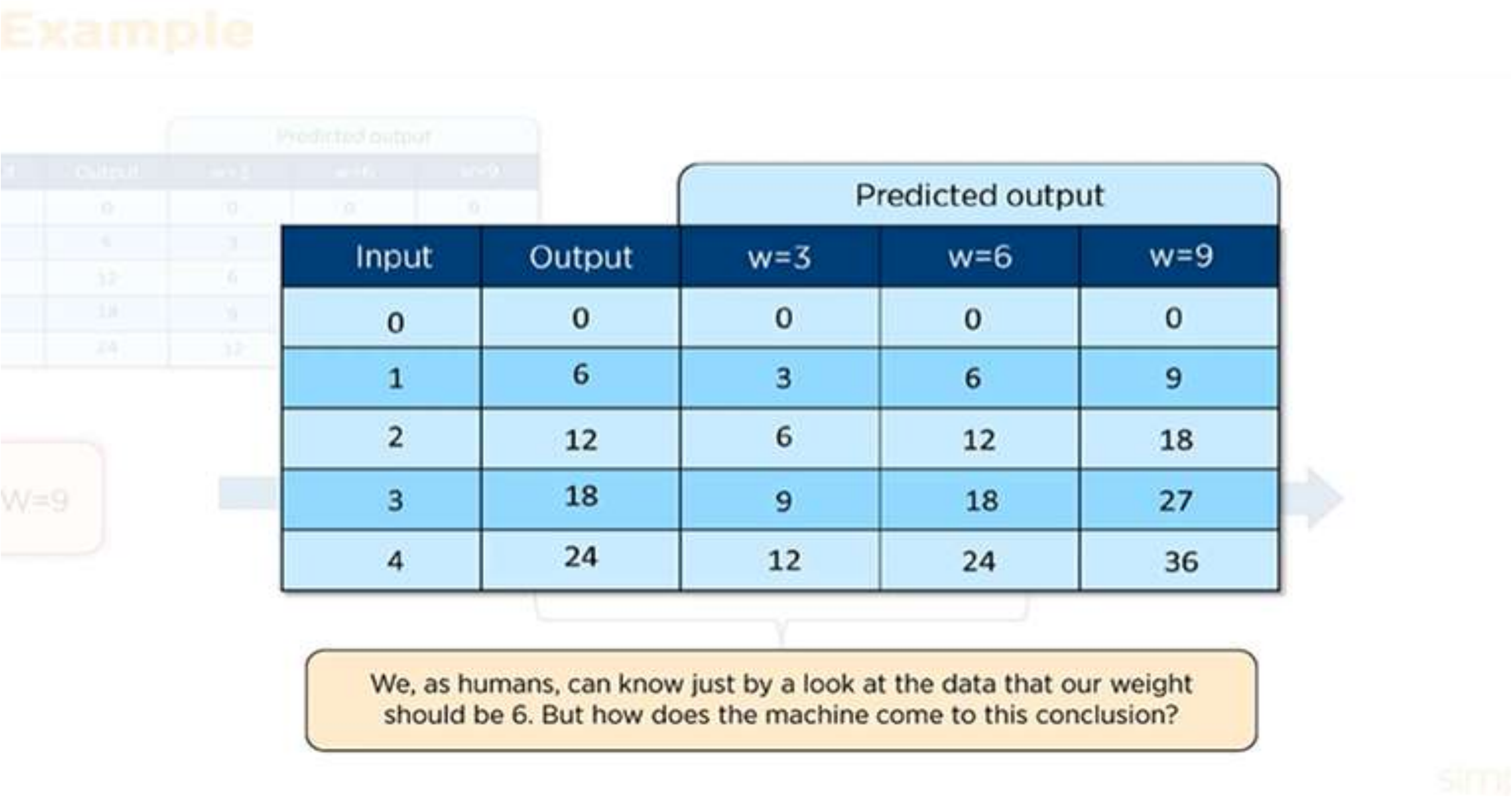


Example

Input	Output
0	0
1	6
2	12
3	18
4	24

The network starts training itself by choosing a random value for w
Input
x
x*w
Output
y
Neural Network

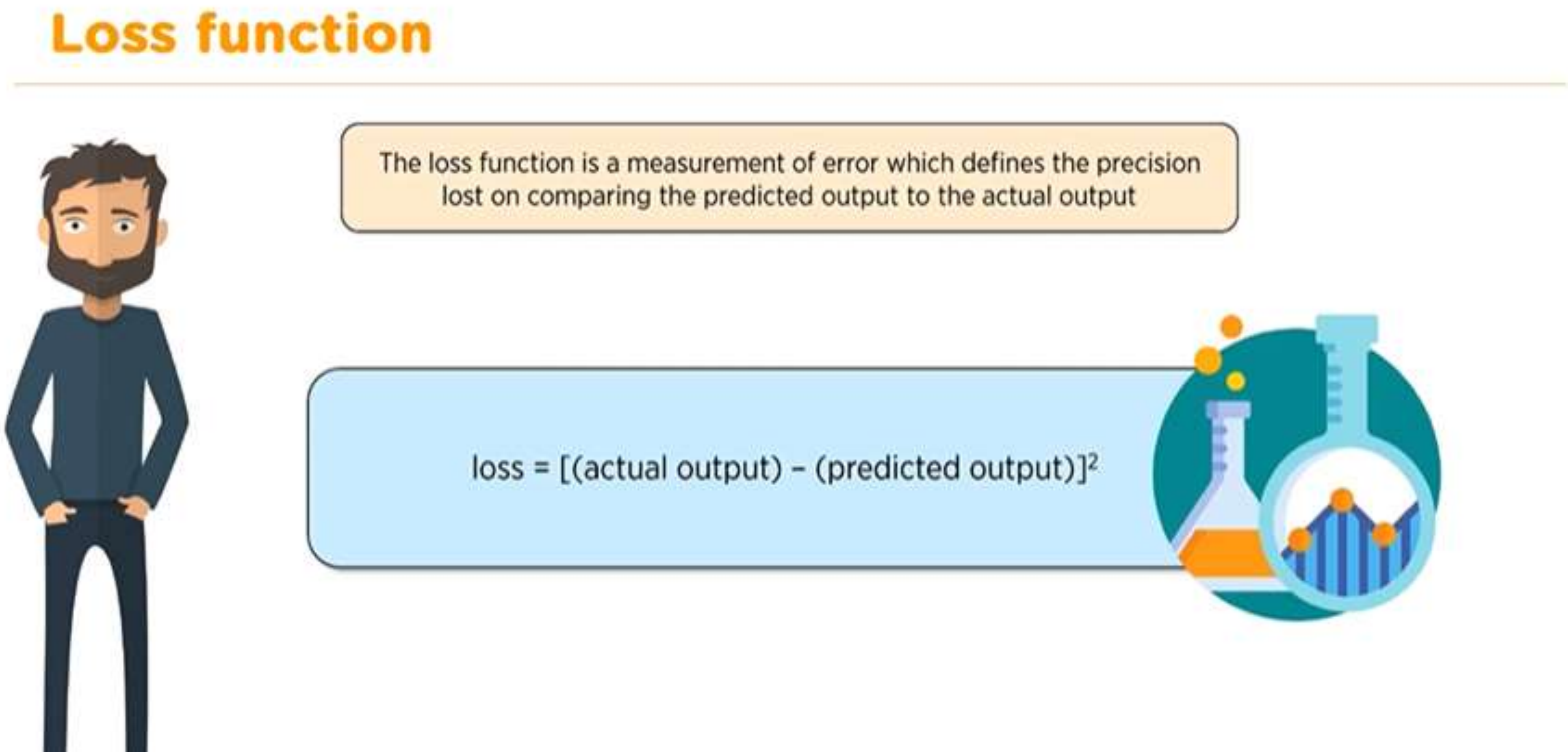
TRAINING A NETWORK



		Predicted output		
Input	Output	w=3	w=6	w=9
0	0	0	0	0
1	6	3	6	9
2	12	6	12	18
3	18	9	18	27
4	24	12	24	36

We, as humans, can know just by a look at the data that our weight should be 6. But how does the machine come to this conclusion?

TRAINING A NETWORK

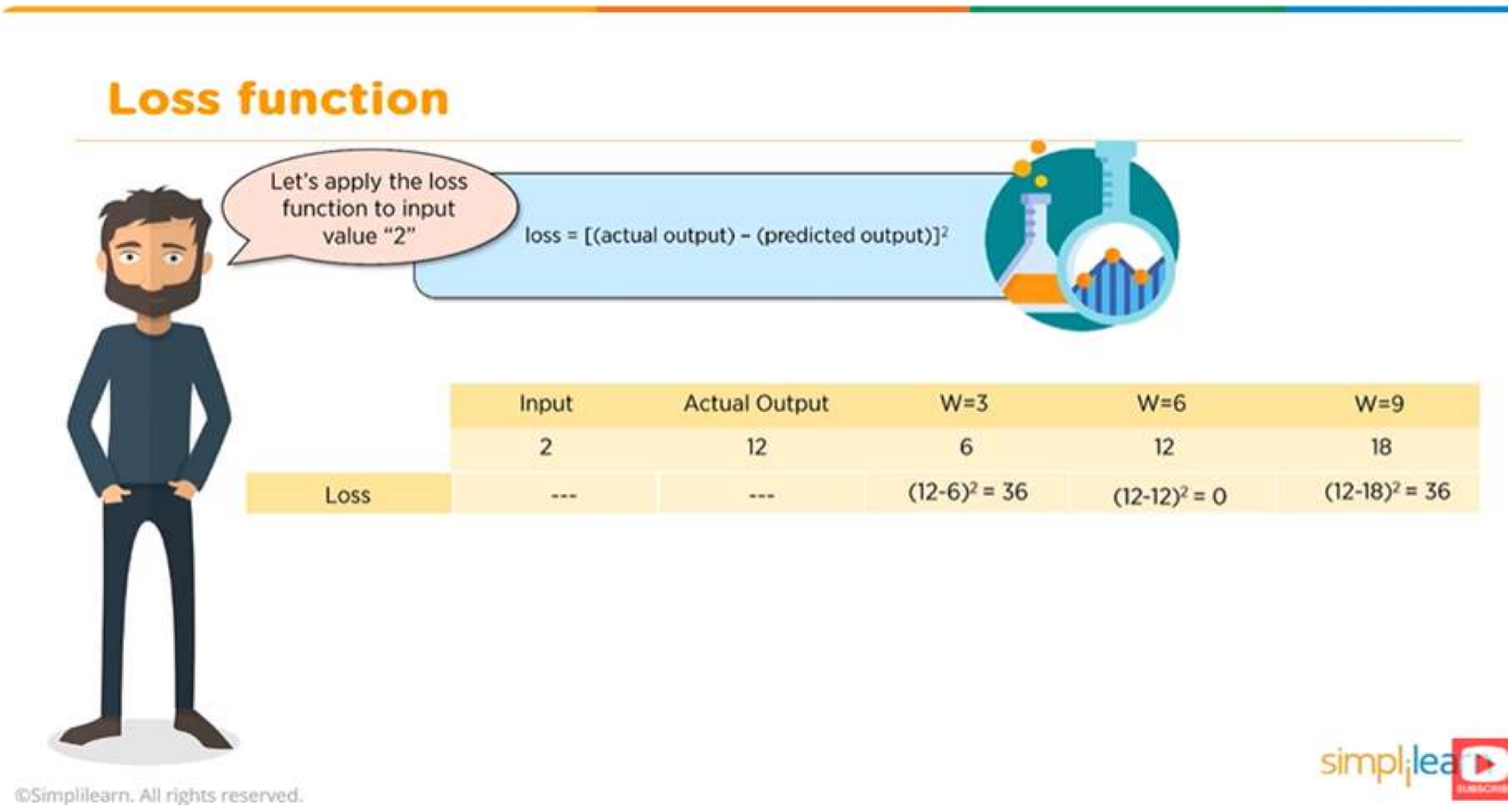


Loss function

The loss function is a measurement of error which defines the precision lost on comparing the predicted output to the actual output

loss = [(actual output) - (predicted output)]^2

TRAINING A NETWORK



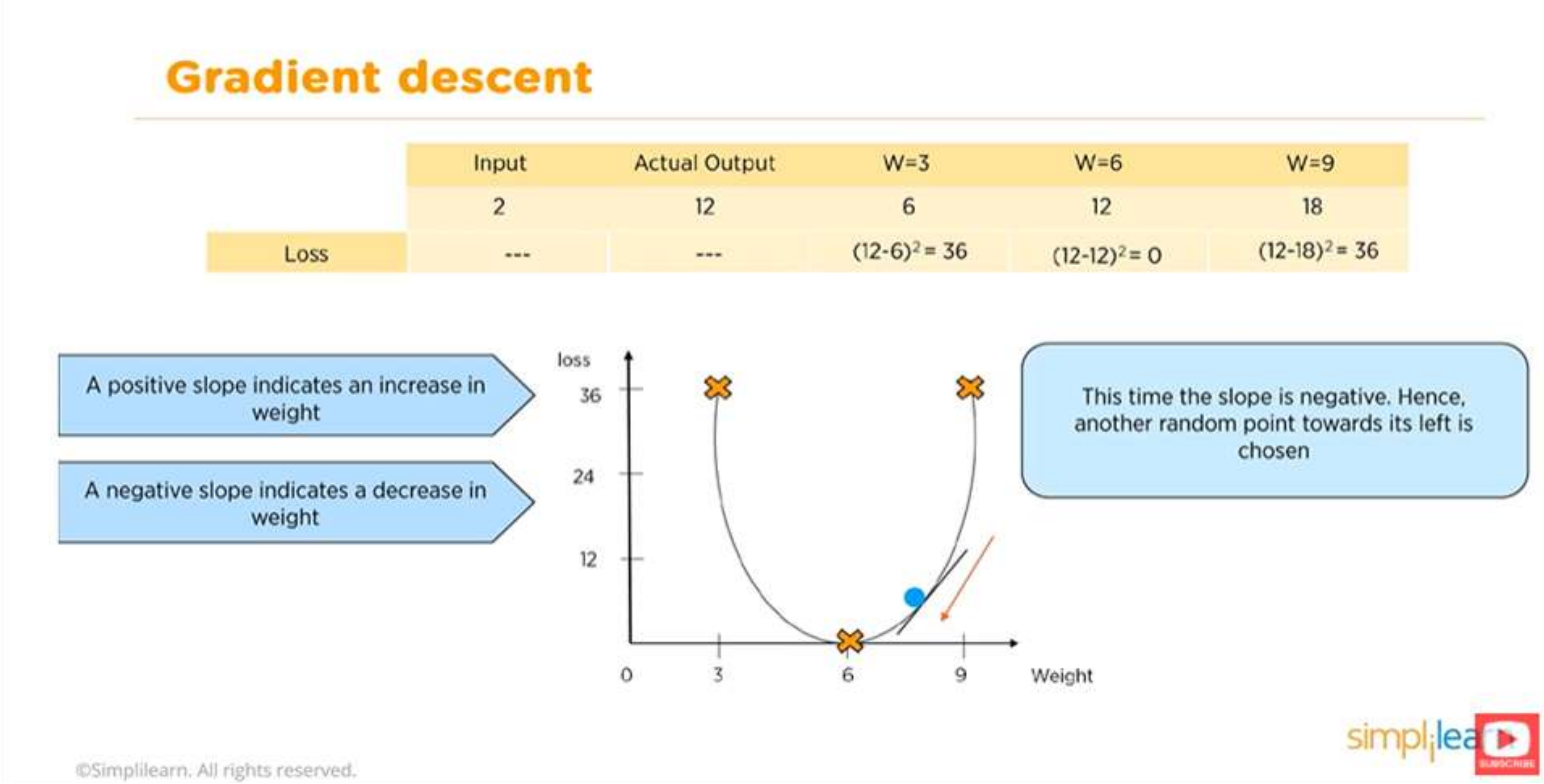
Loss function

Let's apply the loss function to input value "2"

loss = [(actual output) – (predicted output)]^2

	Input	Actual Output	W=3	W=6	W=9
	2	12	6	12	18
Loss	---	---	$(12-6)^2 = 36$	$(12-12)^2 = 0$	$(12-18)^2 = 36$

TRAINING A NETWORK



Gradient descent

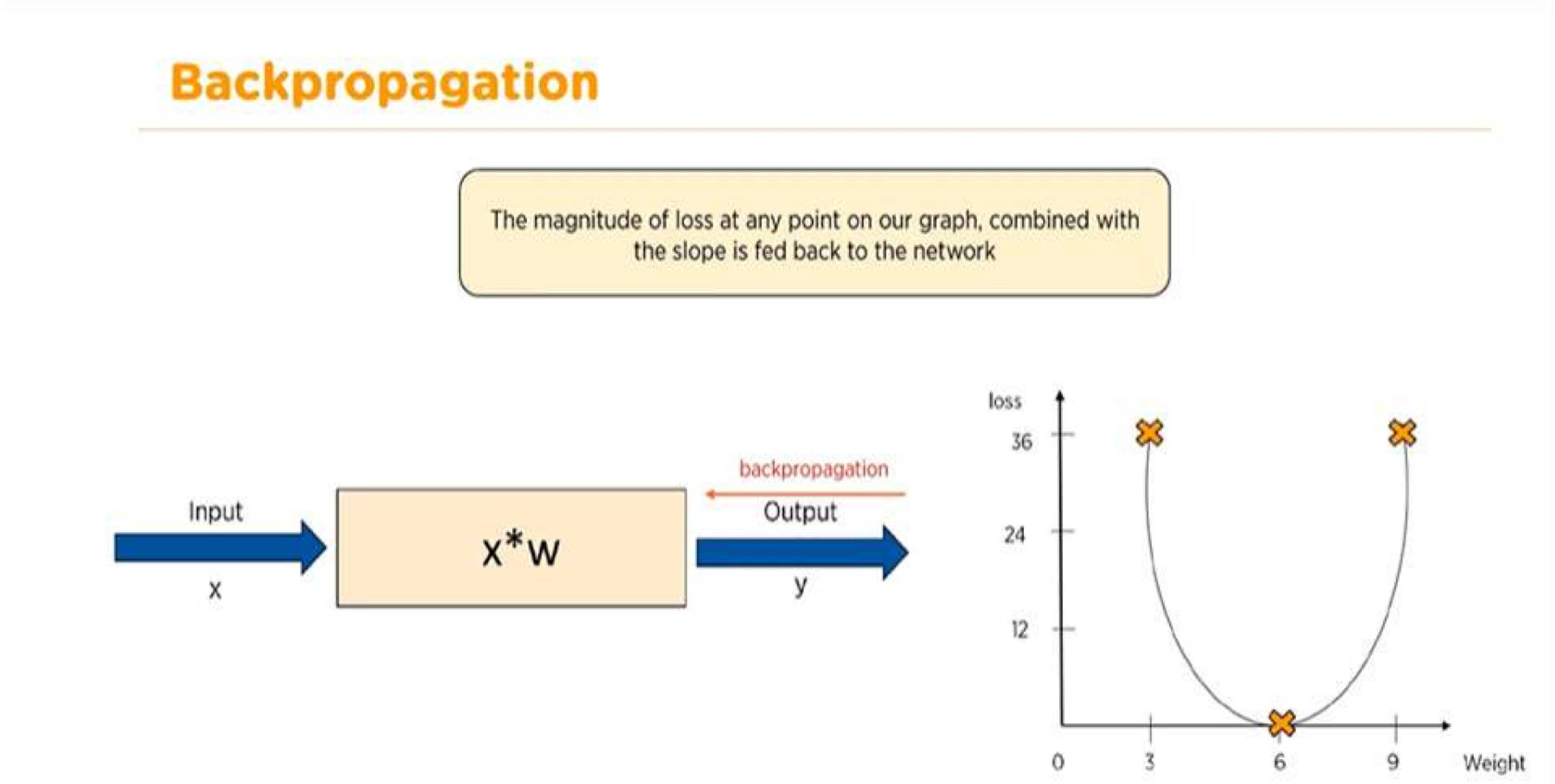
	Input	Actual Output	W=3	W=6	W=9
	2	12	6	12	18
Loss	---	---	$(12-6)^2 = 36$	$(12-12)^2 = 0$	$(12-18)^2 = 36$

A positive slope indicates an increase in weight

A negative slope indicates a decrease in weight

This time the slope is negative. Hence, another random point towards its left is chosen

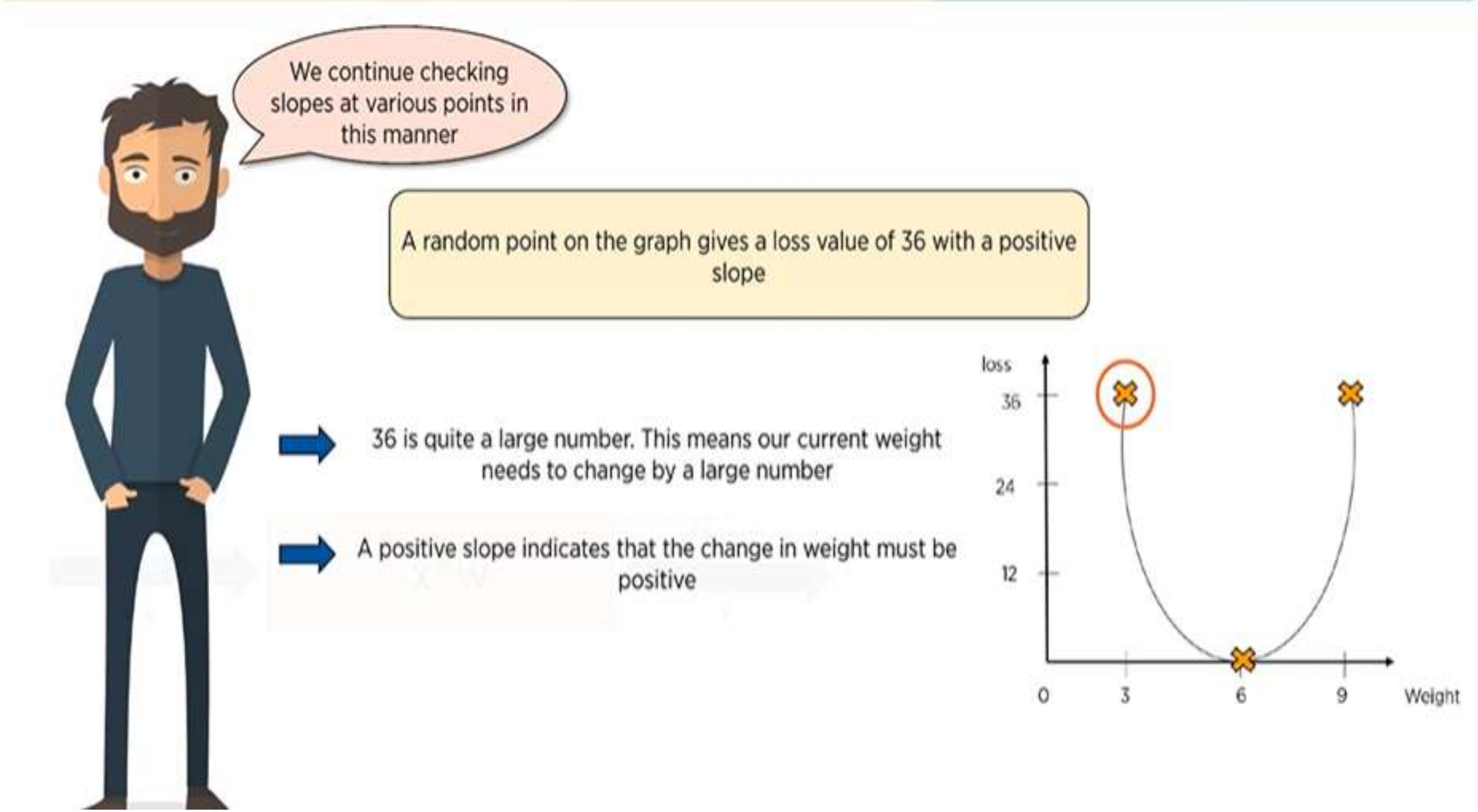
TRAINING A NETWORK



Backpropagation

The magnitude of loss at any point on our graph, combined with the slope is fed back to the network

TRAINING A NETWORK



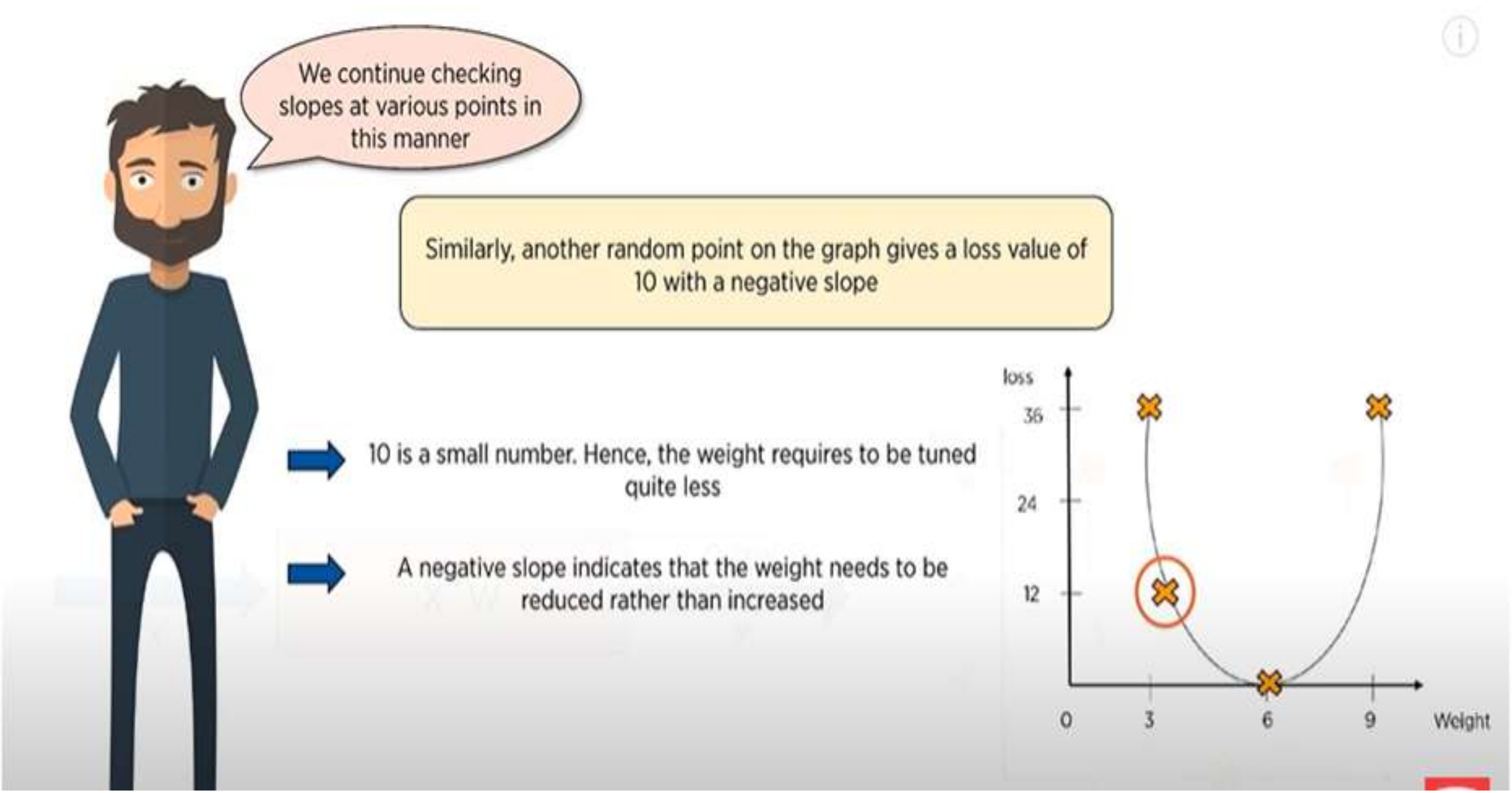
We continue checking slopes at various points in this manner

A random point on the graph gives a loss value of 36 with a positive slope

36 is quite a large number. This means our current weight needs to change by a large number

A positive slope indicates that the change in weight must be positive

TRAINING A NETWORK



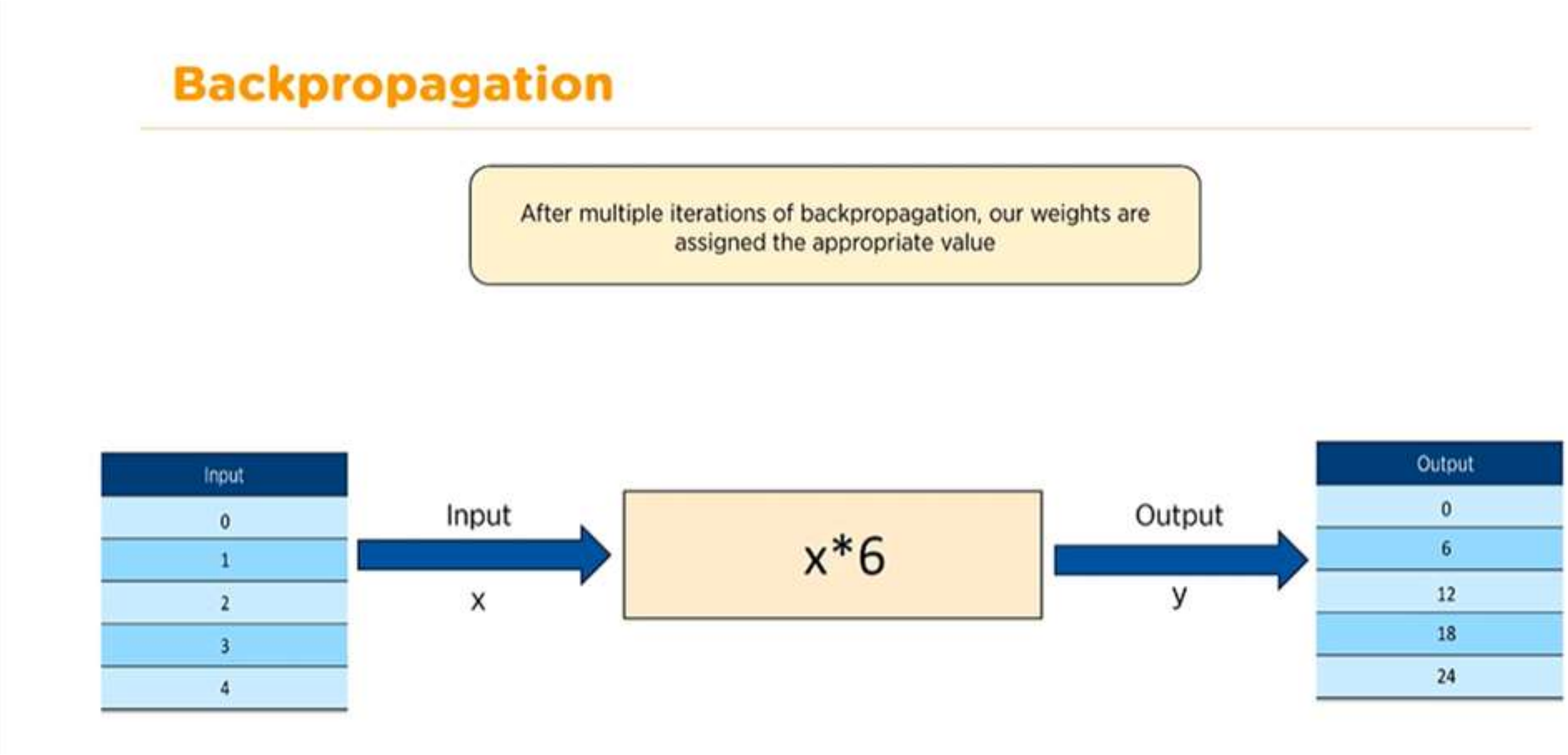
We continue checking slopes at various points in this manner

Similarly, another random point on the graph gives a loss value of 10 with a negative slope

10 is a small number. Hence, the weight requires to be tuned quite less

A negative slope indicates that the weight needs to be reduced rather than increased

TRAINING A NETWORK



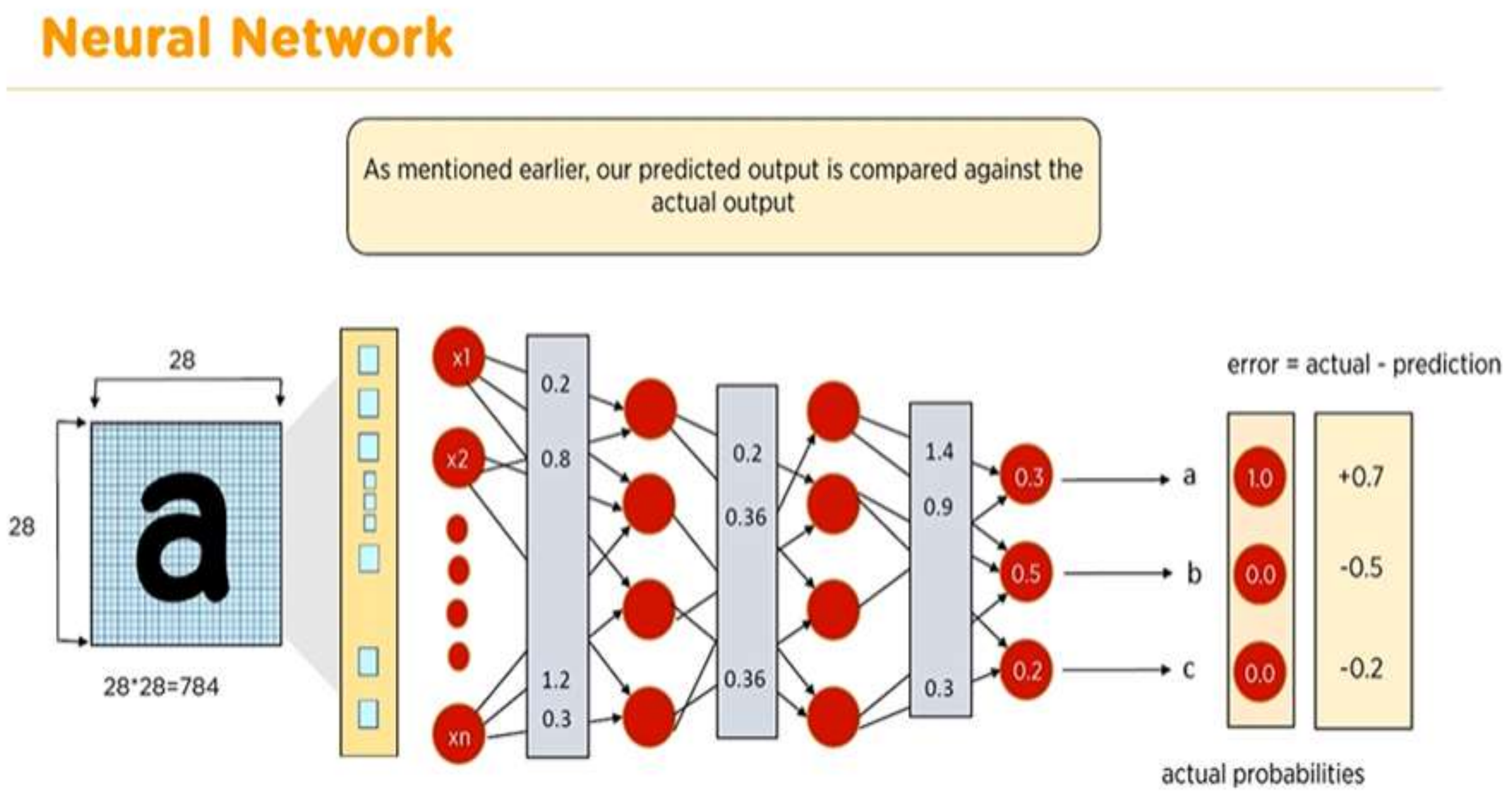
Backpropagation

After multiple iterations of backpropagation, our weights are assigned the appropriate value

Input
0
1
2
3
4

Output
0
6
12
18
24

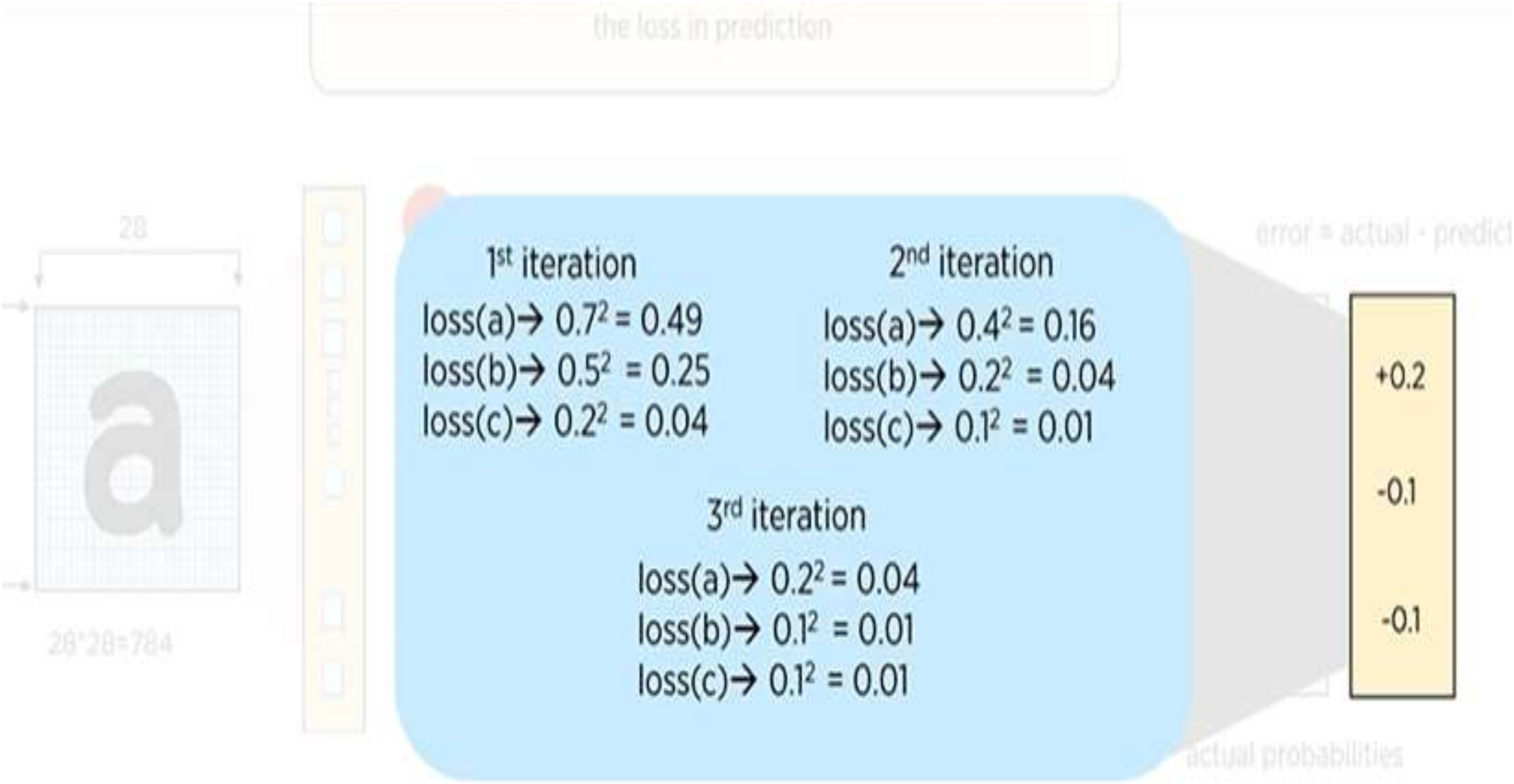
TRAINING A NETWORK



Neural Network

As mentioned earlier, our predicted output is compared against the actual output

TRAINING A NETWORK



1st iteration

loss(a)→ $0.7^2 = 0.49$
loss(b)→ $0.5^2 = 0.25$
loss(c)→ $0.2^2 = 0.04$

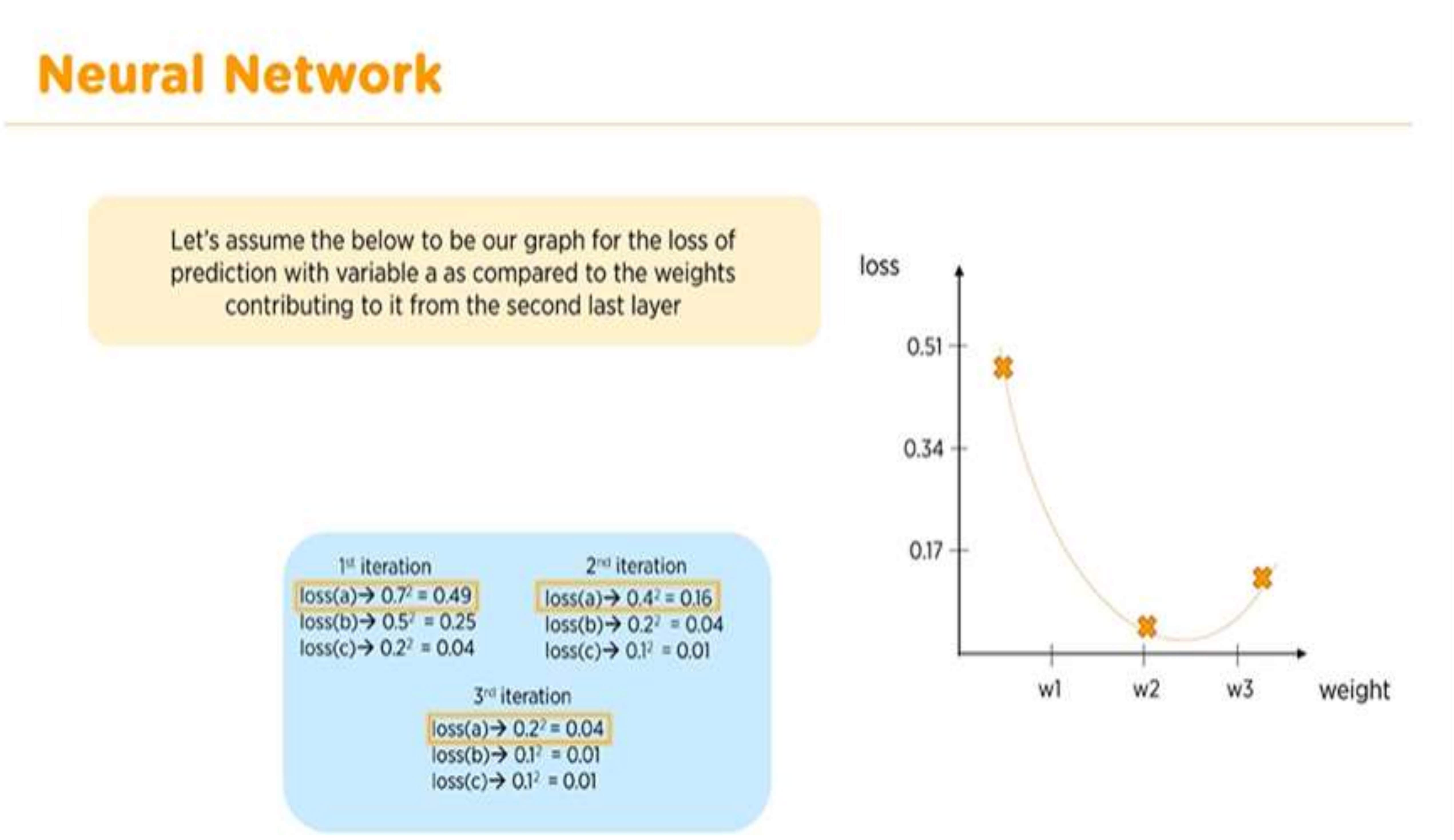
2nd iteration

loss(a)→ $0.4^2 = 0.16$
loss(b)→ $0.2^2 = 0.04$
loss(c)→ $0.1^2 = 0.01$

3rd iteration

loss(a)→ $0.2^2 = 0.04$
loss(b)→ $0.1^2 = 0.01$
loss(c)→ $0.1^2 = 0.01$

TRAINING A NETWORK



Neural Network

Let's assume the below to be our graph for the loss of prediction with variable a as compared to the weights contributing to it from the second last layer

1st iteration

loss(a)→ $0.7^2 = 0.49$

loss(b)→ $0.5^2 = 0.25$

loss(c)→ $0.2^2 = 0.04$

2nd iteration

loss(a)→ $0.4^2 = 0.16$

loss(b)→ $0.2^2 = 0.04$

loss(c)→ $0.1^2 = 0.01$

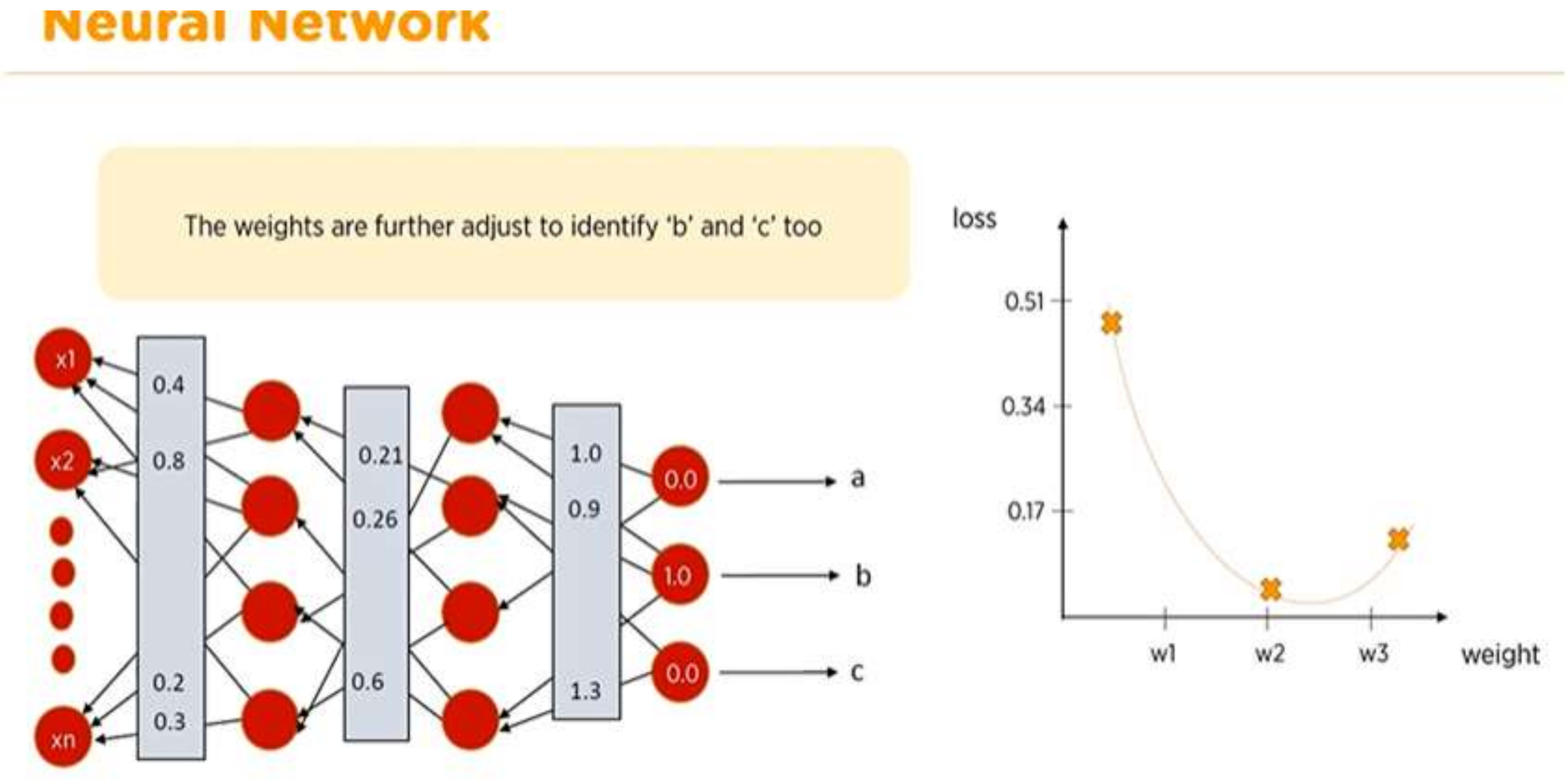
3rd iteration

loss(a)→ $0.2^2 = 0.04$

loss(b)→ $0.1^2 = 0.01$

loss(c)→ $0.1^2 = 0.01$

TRAINING A NETWORK



Neural Network

The weights are further adjust to identify ‘b’ and ‘c’ too

TRAINING A NETWORK

Activity

Objective: Understand loss calculation and weight updates manually.

1. Assign random weights and calculate predictions for a small dataset; compute the loss (e.g., MSE).
2. Use backpropagation rules to adjust the weights by subtracting gradients.
3. Repeat for a few epochs and observe how the loss decreases over iterations.

TRAINING A NETWORK

THANK YOU