

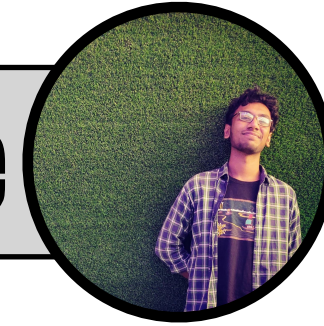
Deep Learning: 101



Dr. Antonio Mastropaolo

Instructor

Mr. Alvi Haque



Teaching Assistant



WILLIAM & MARY

CHARTERED 1693

Spring 2026



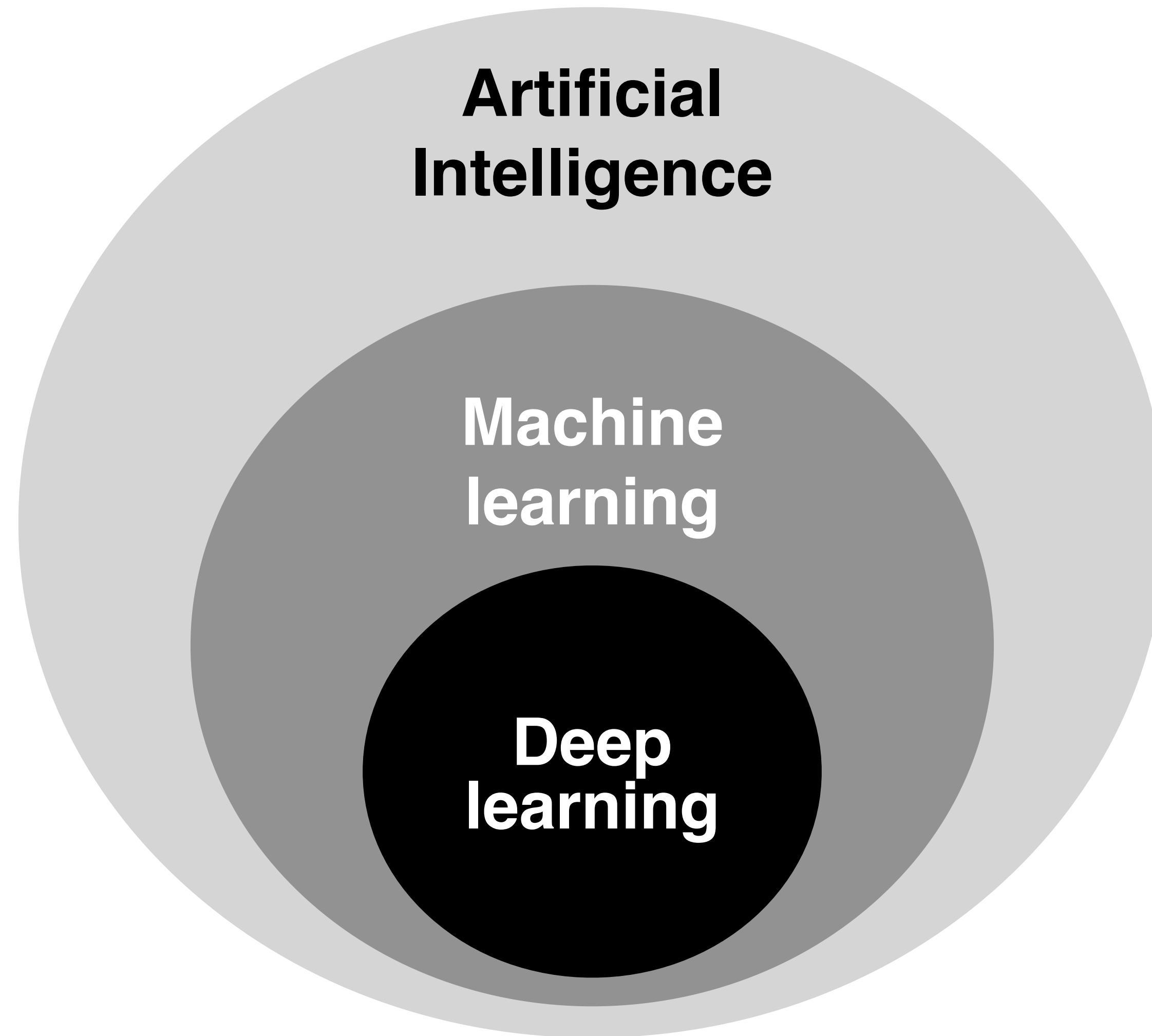
antoniomastropaolo.com



[aura-se-lab.github.io](https://github.com/aura-se-lab)



Deep Learning: 101



Deep Learning: 101

Following slides based on: <https://www.youtube.com/watch?v=aircAruvnKk>



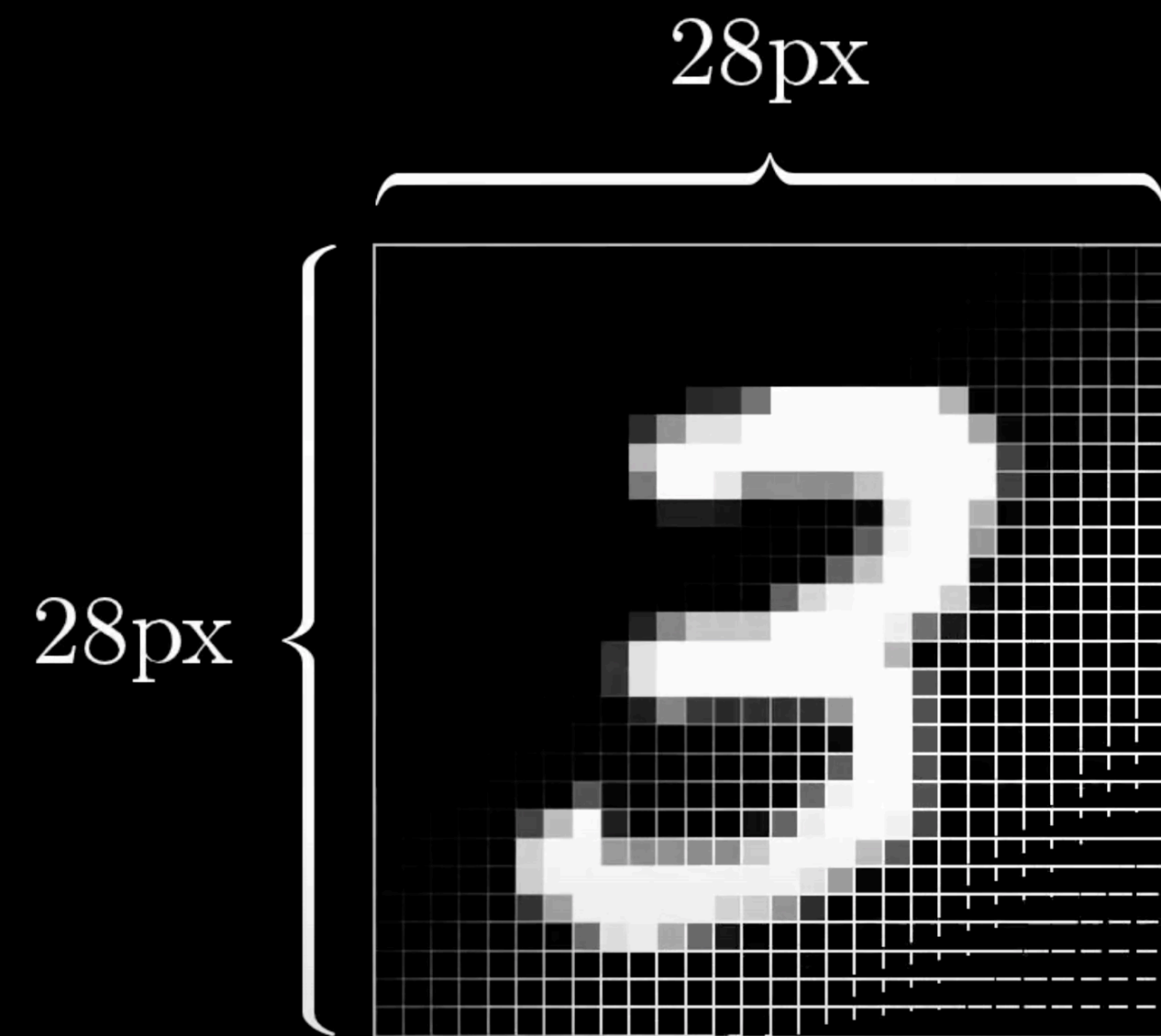
antoniomastropaolo.com



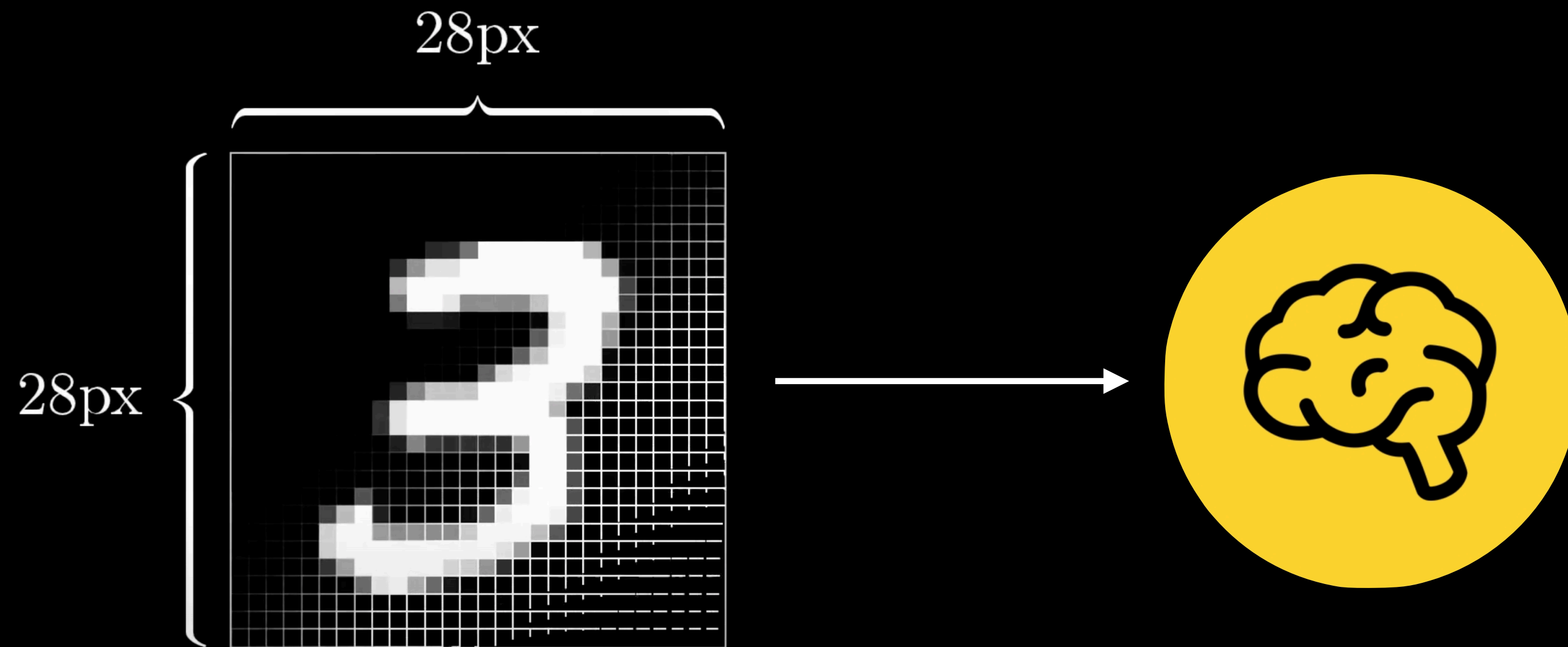
[aura-se-lab.github.io](https://github.com/aura-se-lab)



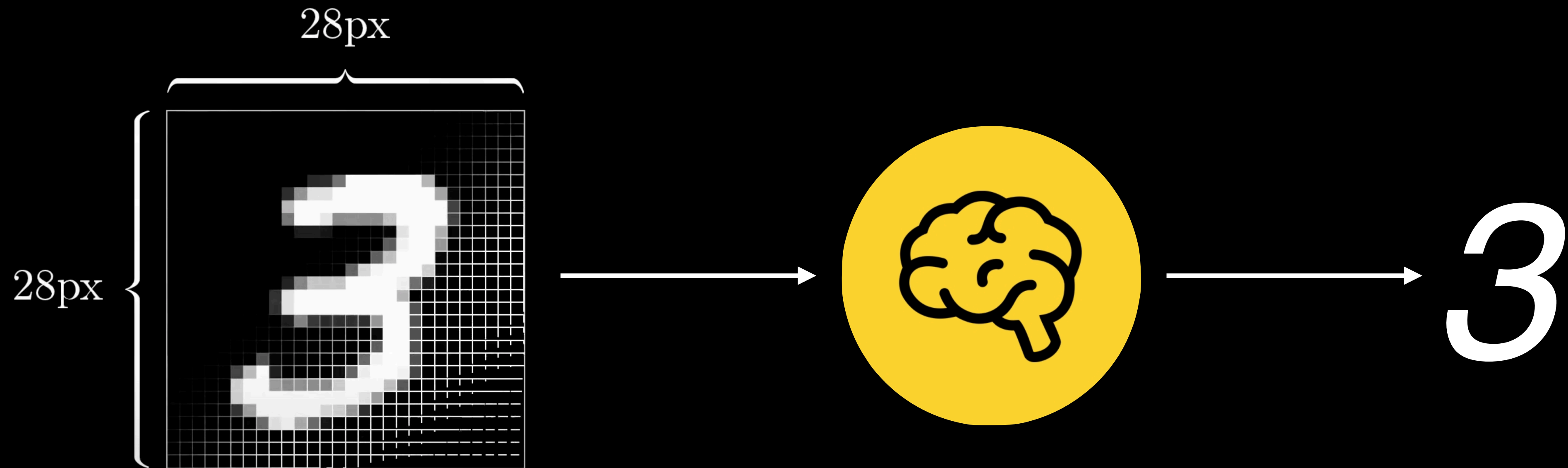
Deep Learning: 101



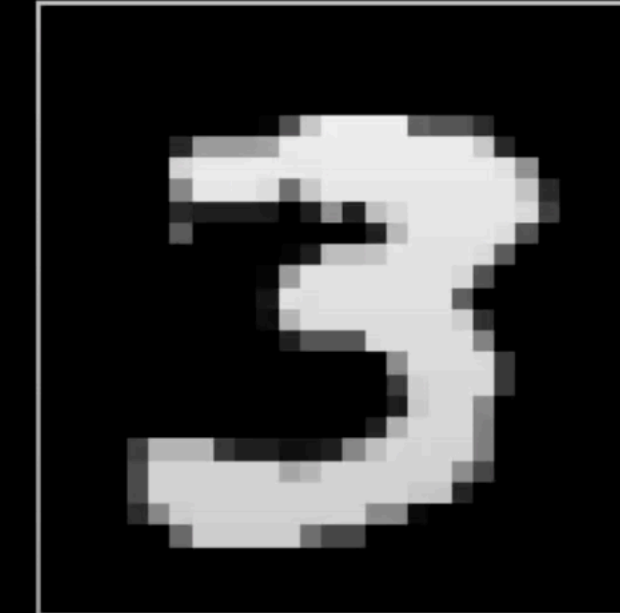
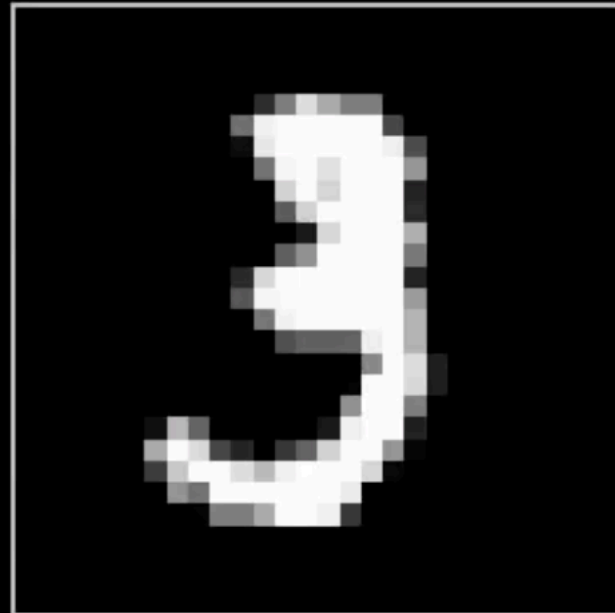
Deep Learning: 101



Deep Learning: 101



Deep Learning: 101

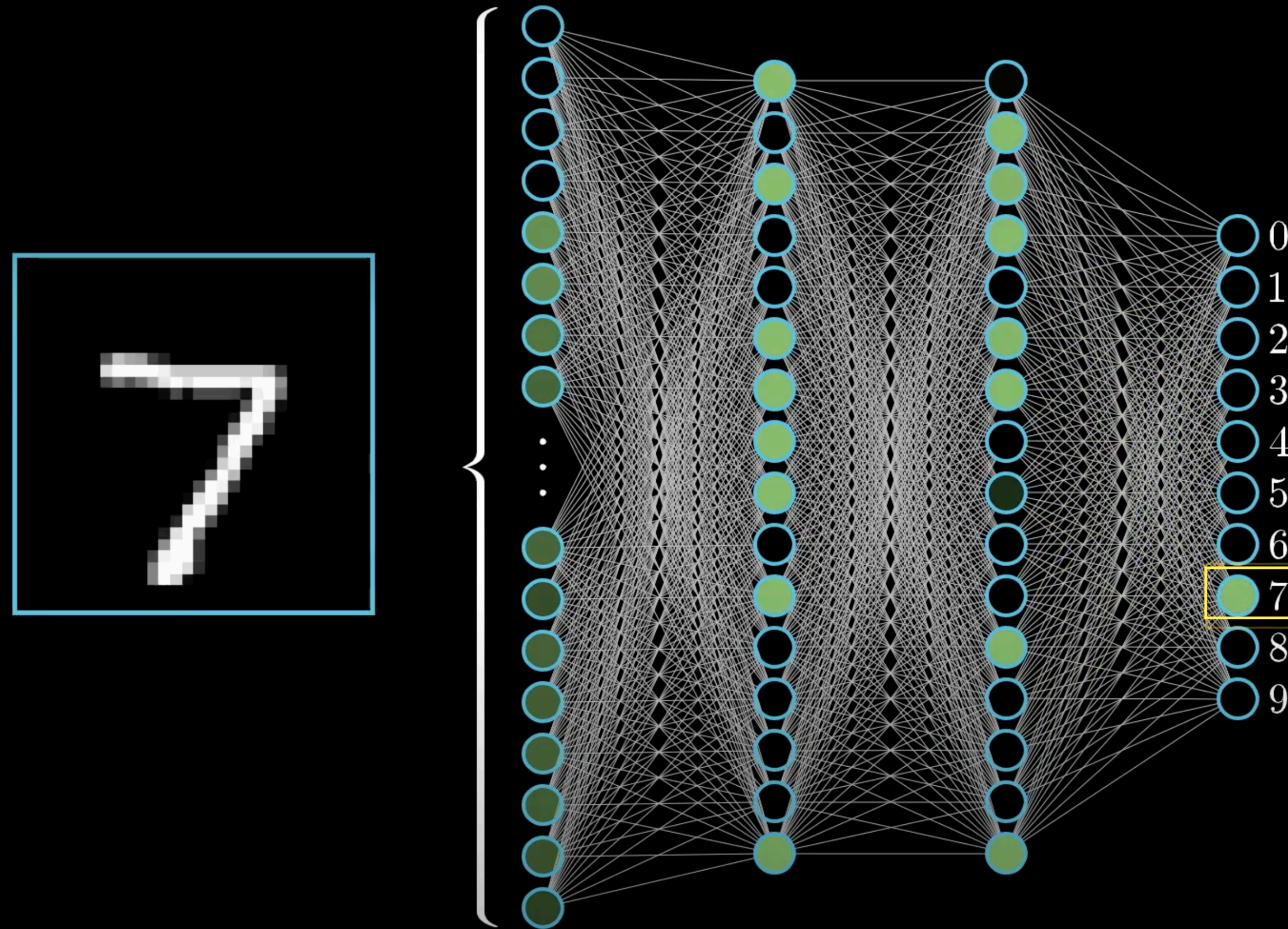


Deep Learning: 101

Can you write a program that, given a 28x28 matrix of pixels as input, recognizes any number between 0 and 9 that is depicted in the image (if any)?



Deep Learning: 101



Deep Learning: 101

Plain vanilla neural network

Deep Learning: 101

0.2

Neuron



antoniomastropaolo.com



[aura-se-lab.github.io](https://github.com/aura-se-lab)



Deep Learning: 101



Neuron

A container that holds a probability



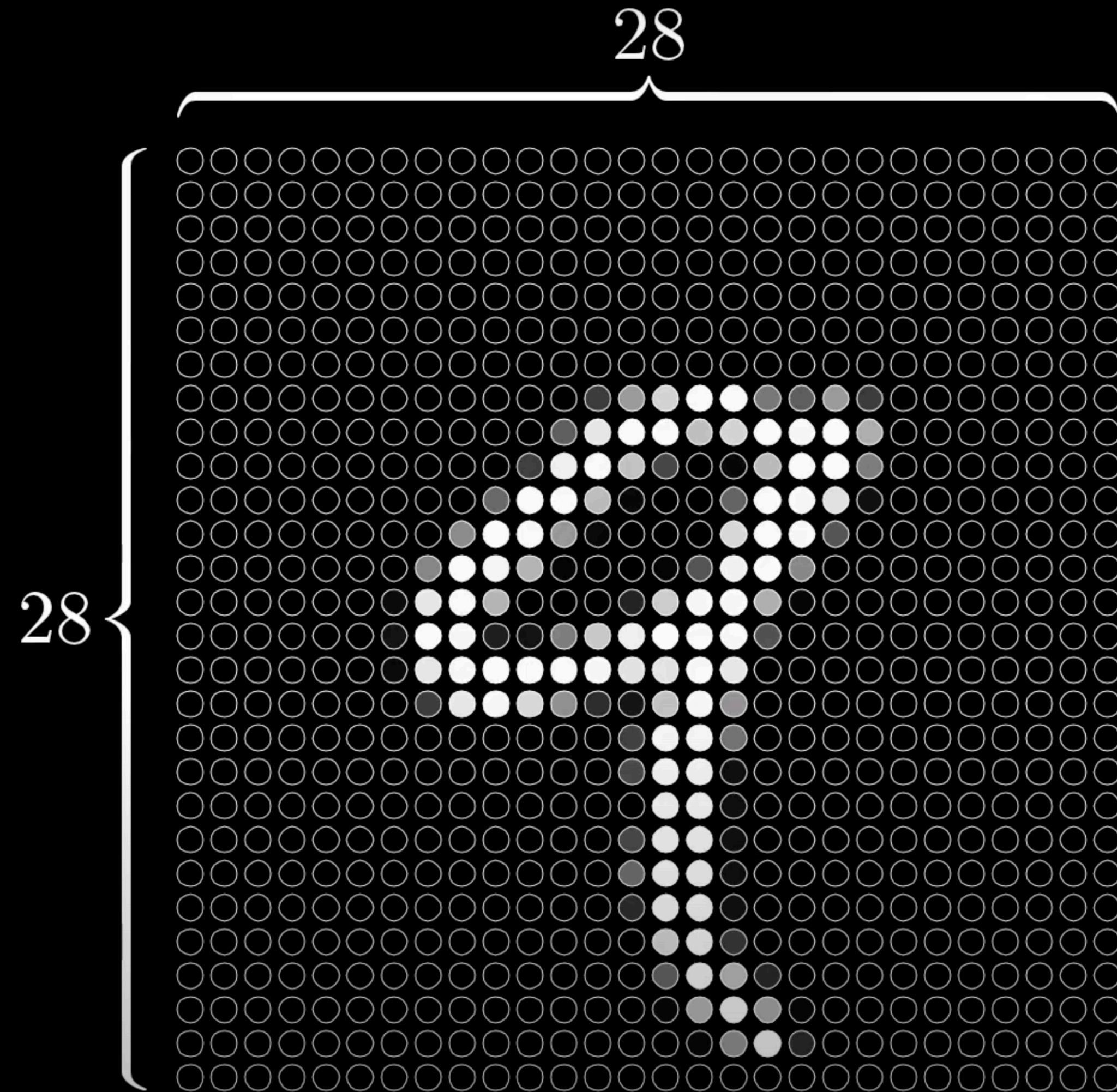
antoniomastropaolo.com



[aura-se-lab.github.io](https://github.com/aura-se-lab)



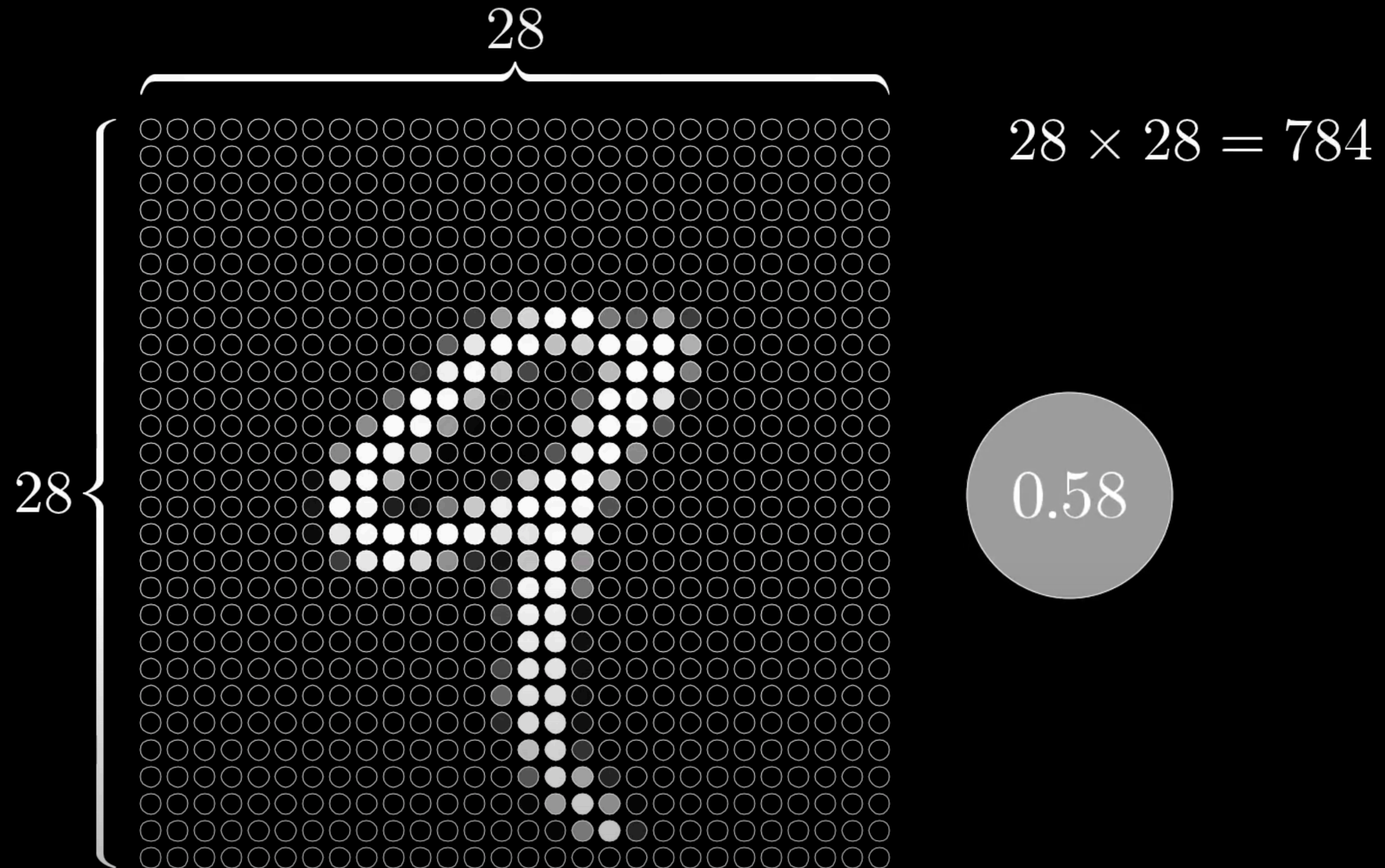
Deep Learning: 101



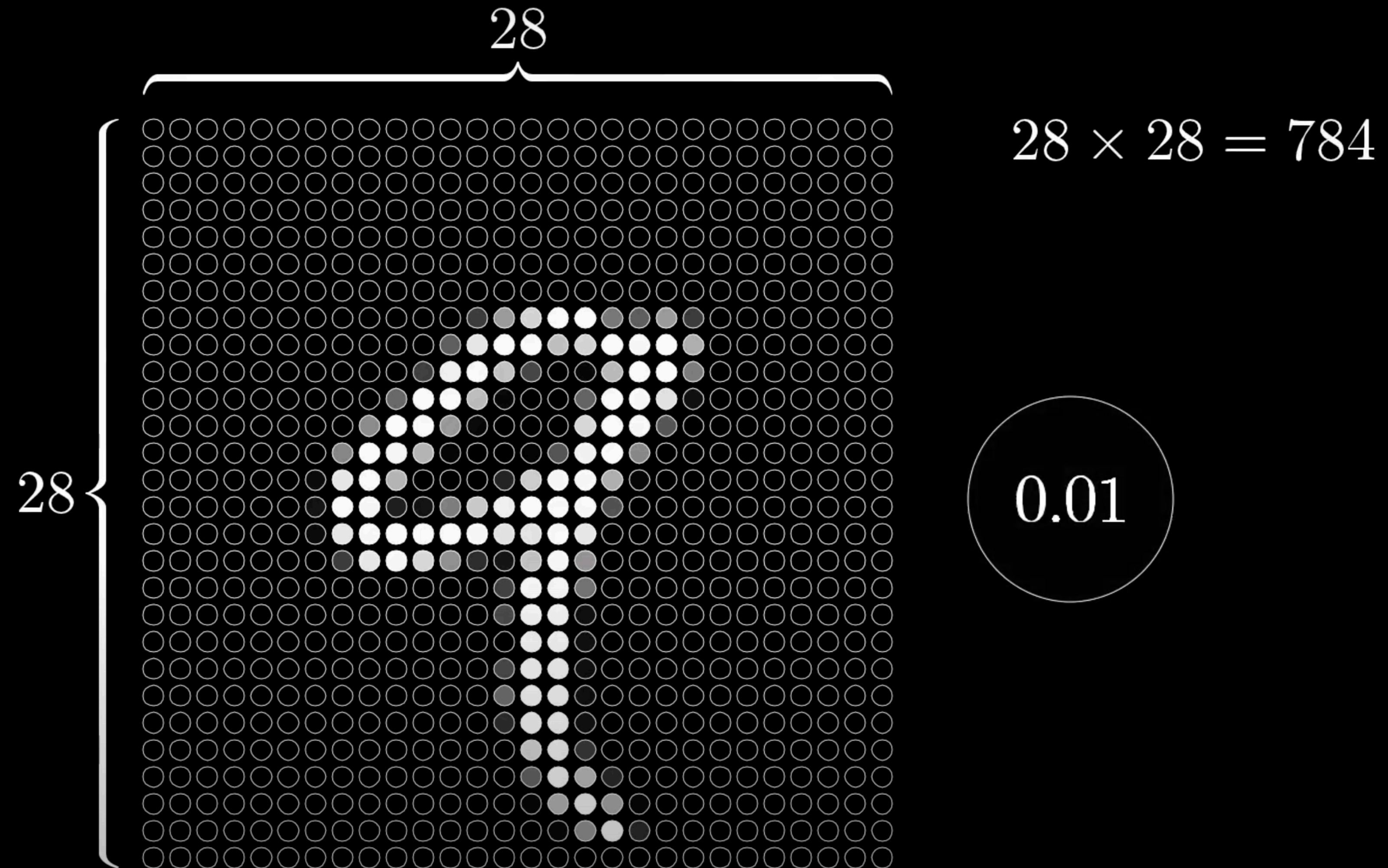
$$28 \times 28 = 784$$



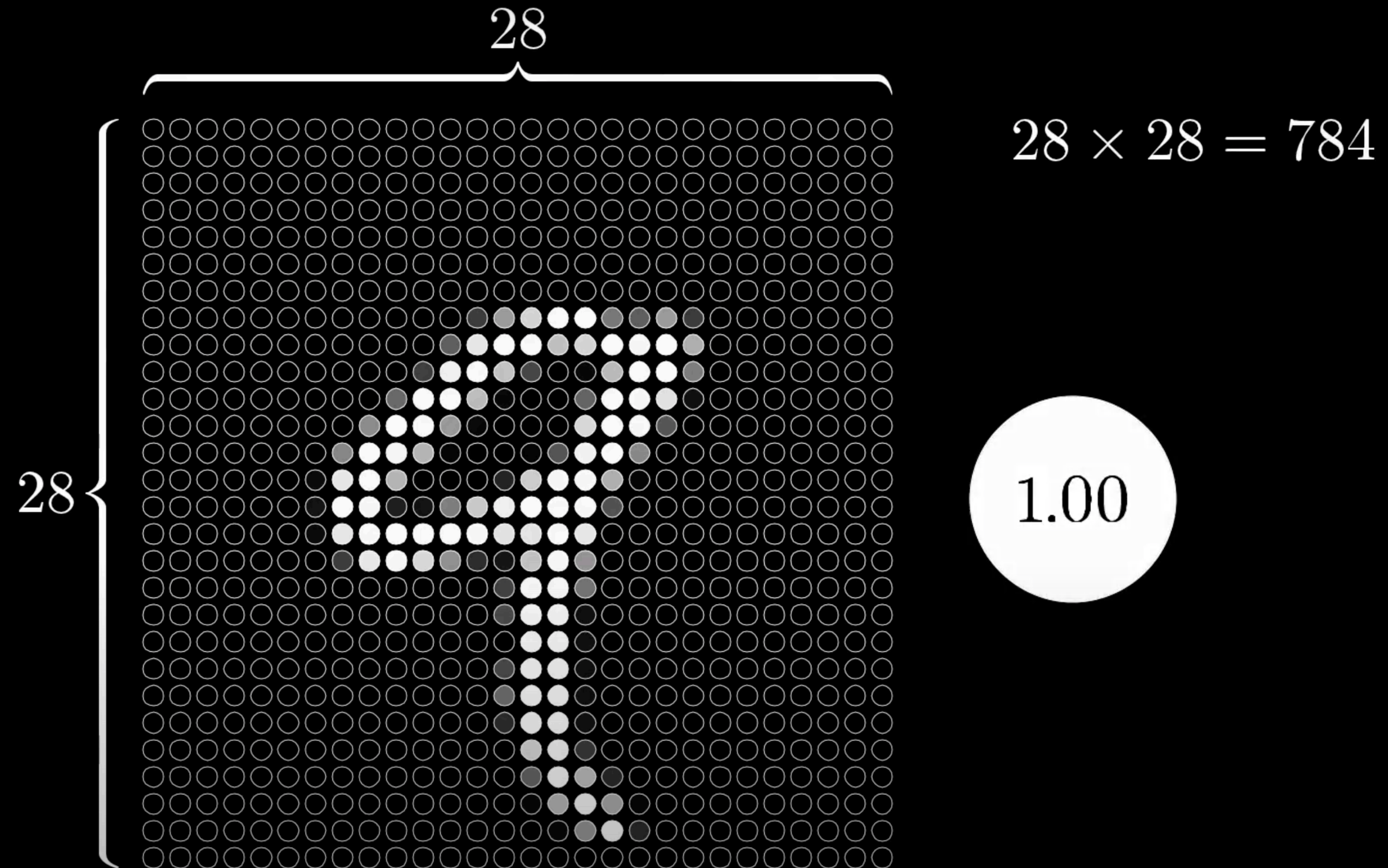
Deep Learning: 101



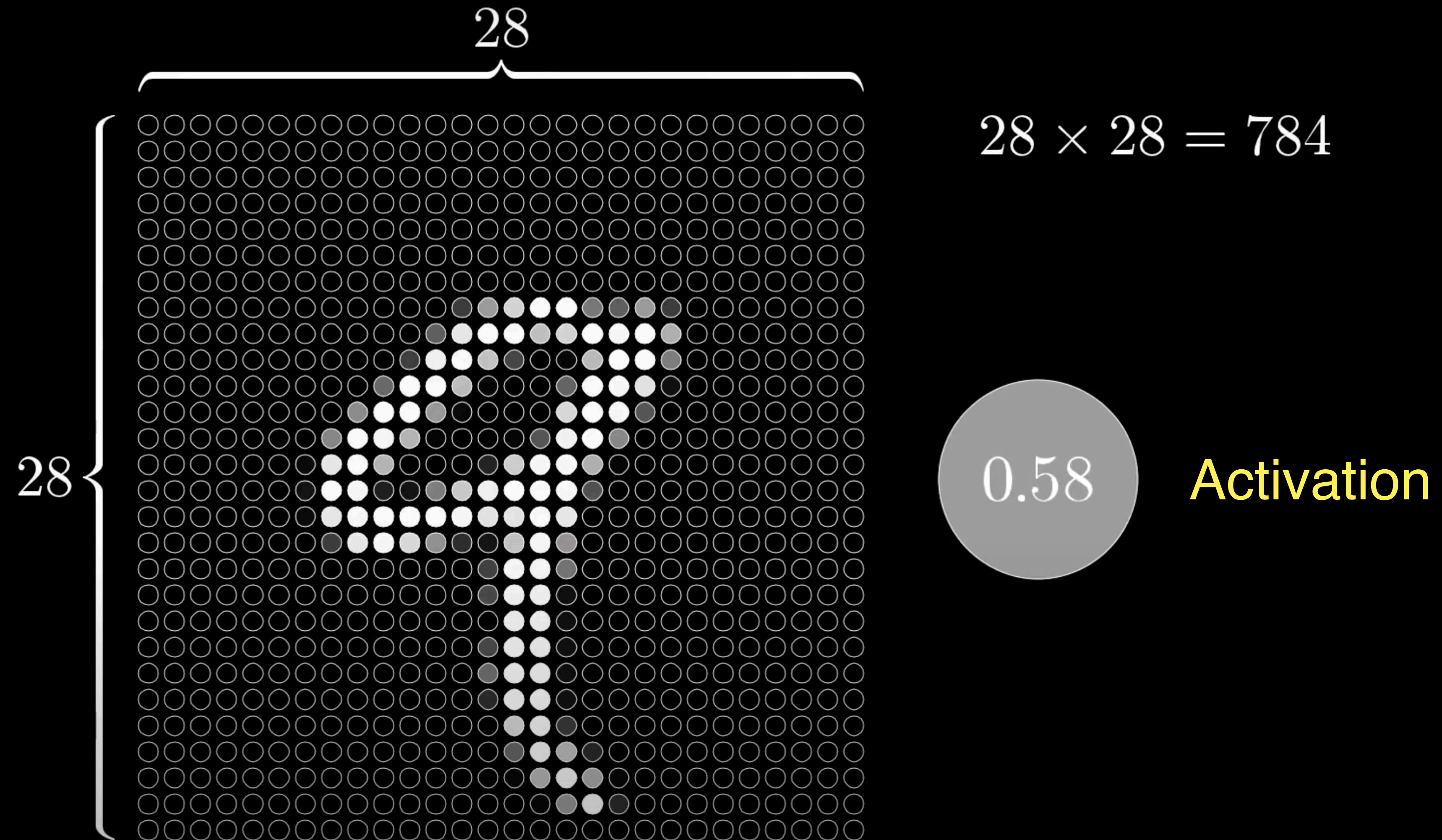
Deep Learning: 101



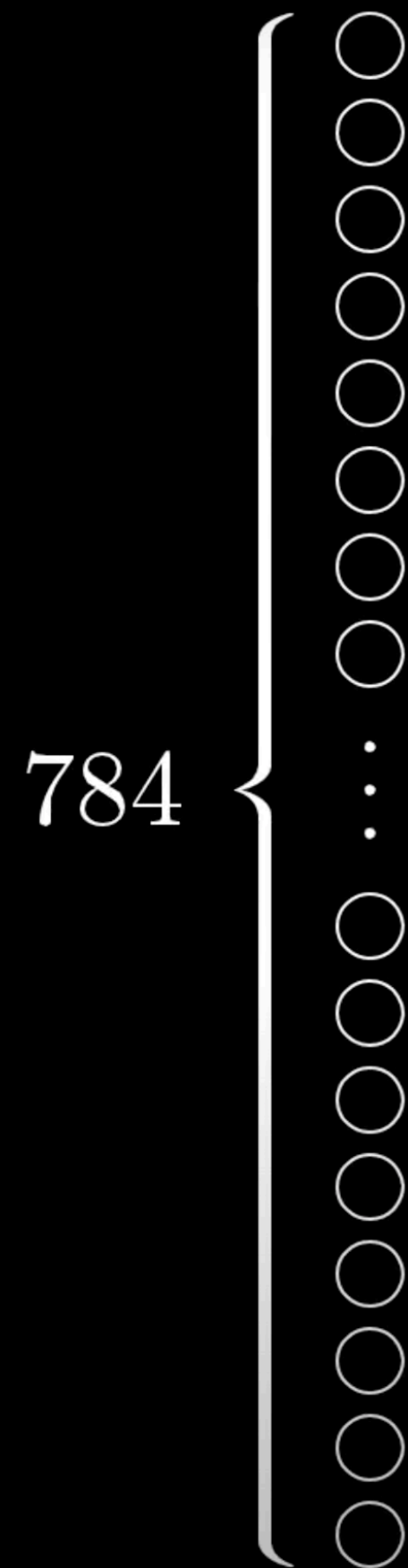
Deep Learning: 101



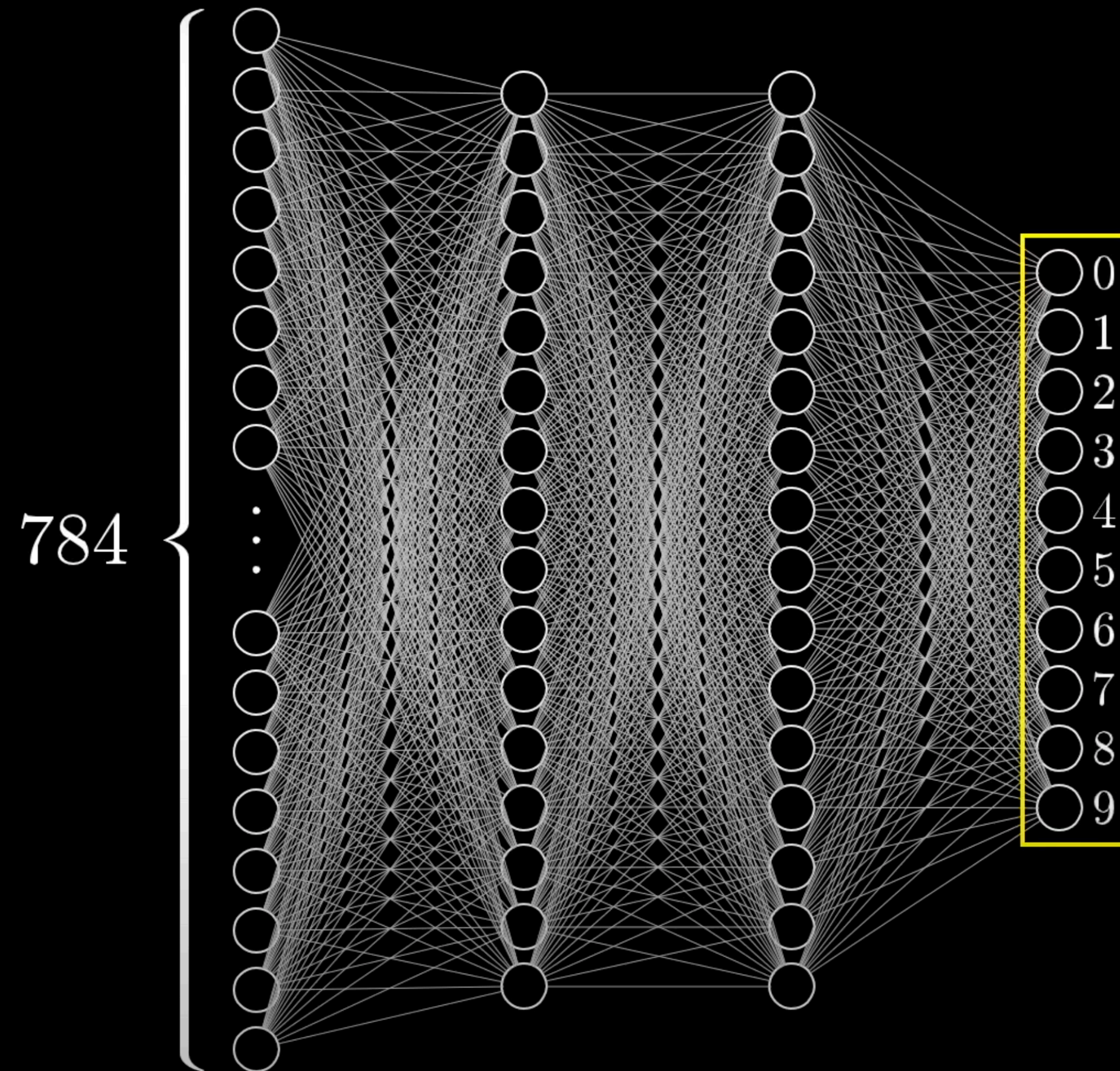
Deep Learning: 101



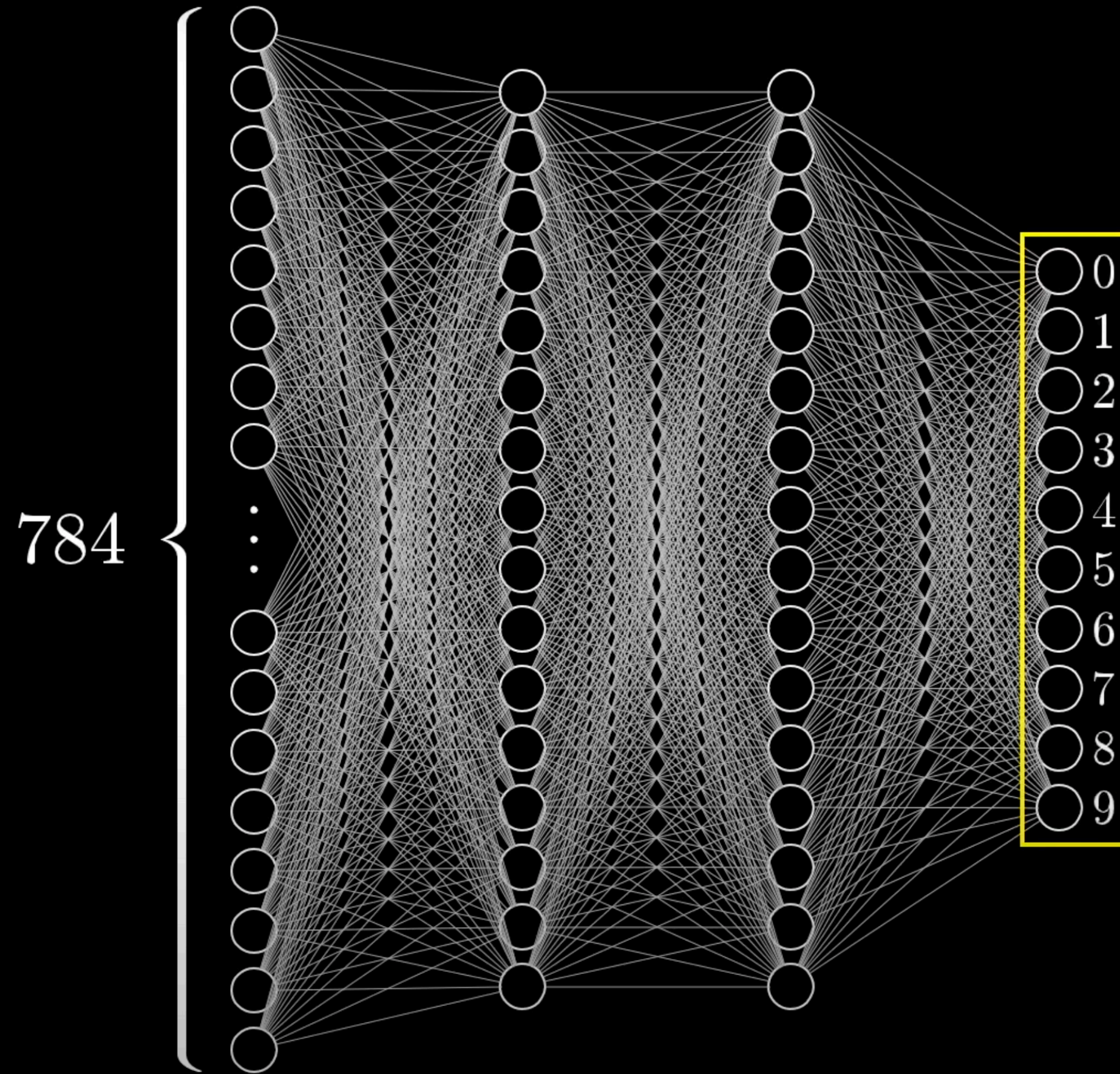
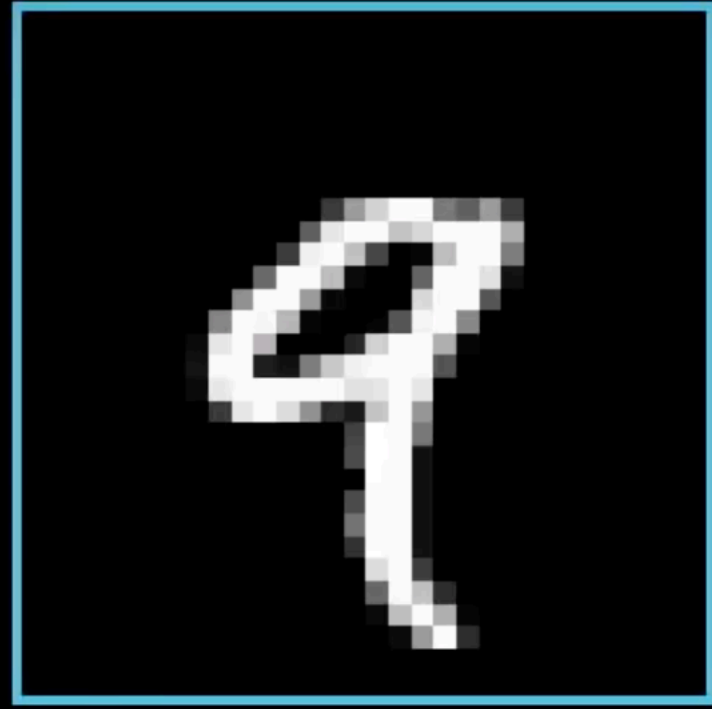
Deep Learning: 101



Deep Learning: 101



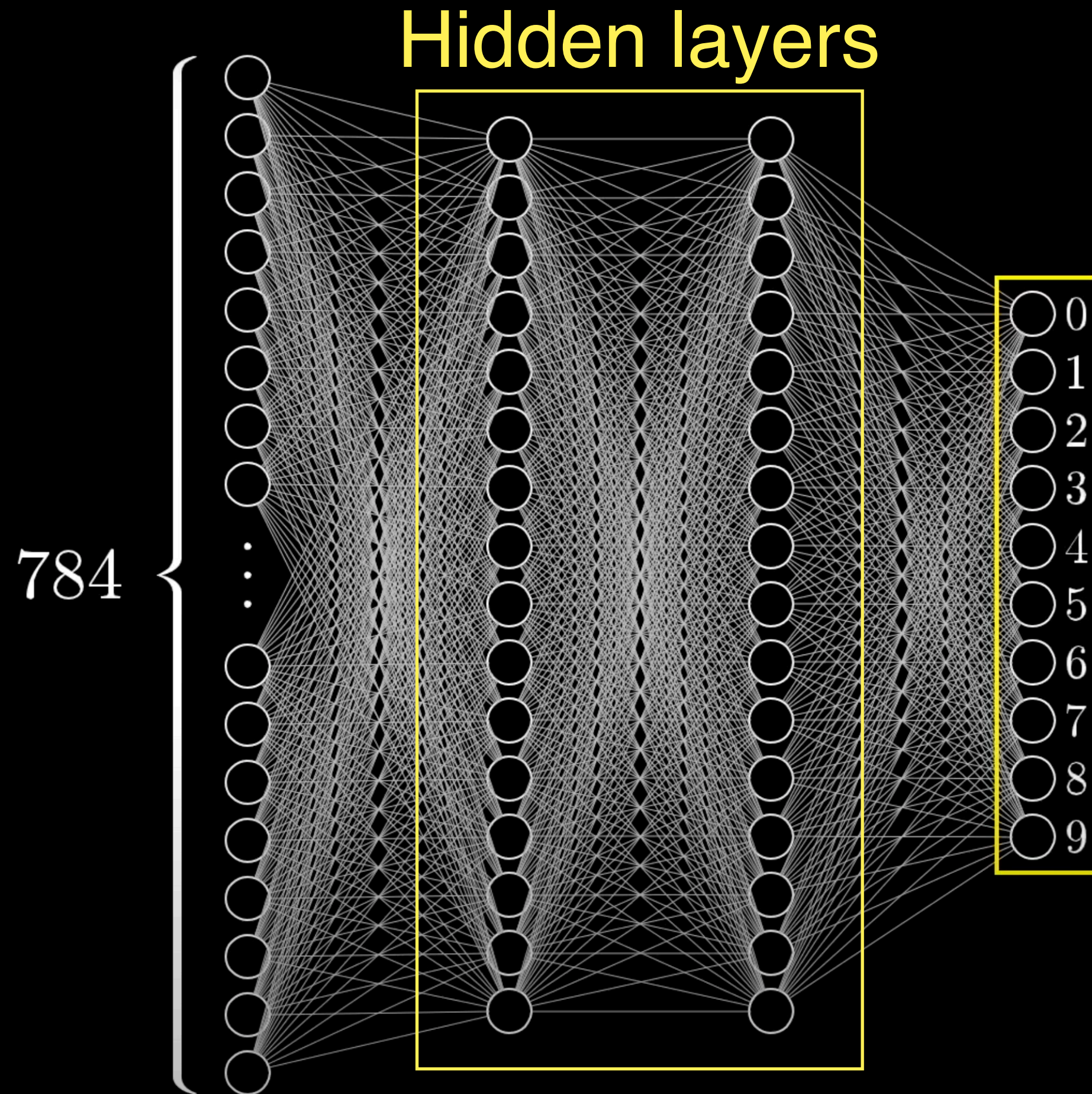
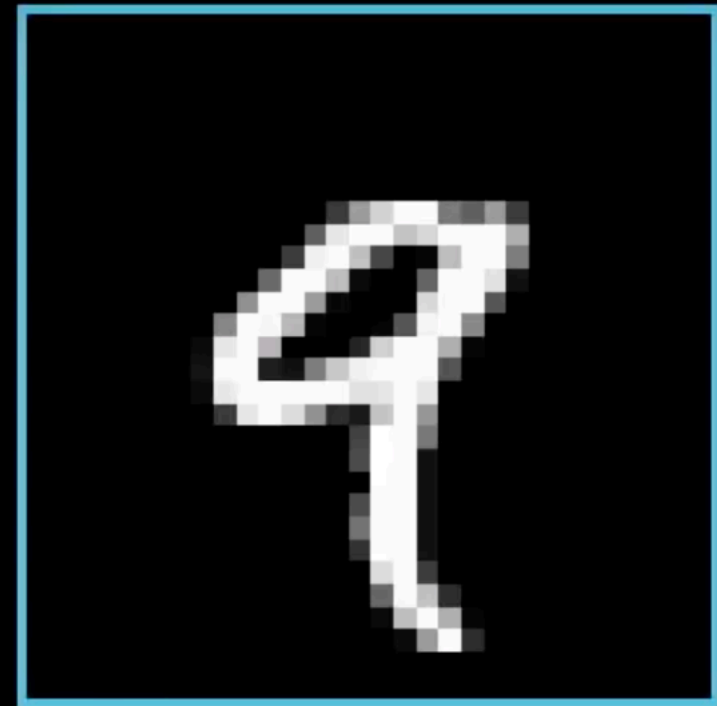
Deep Learning: 101



0.97 9

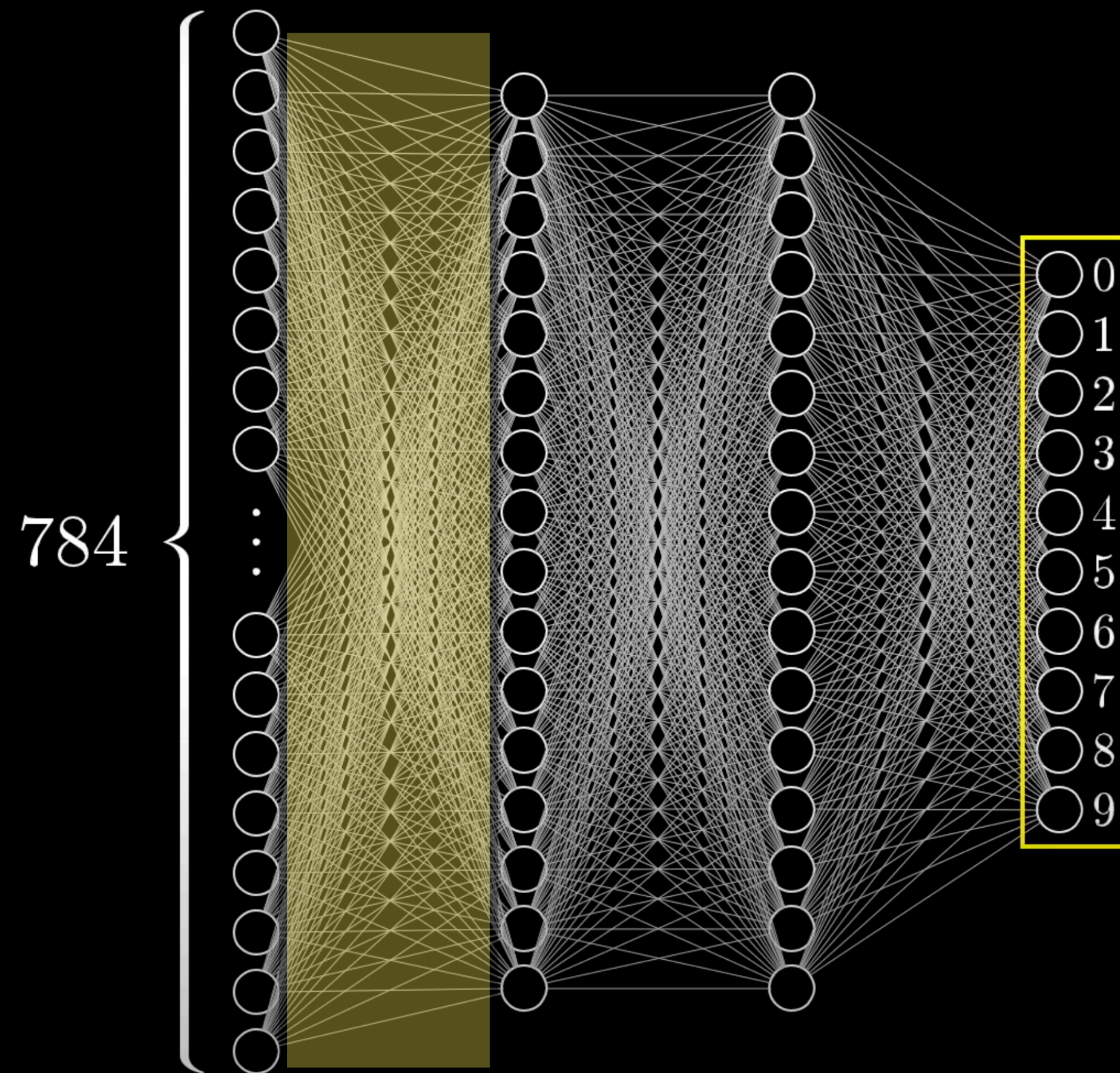


Deep Learning: 101

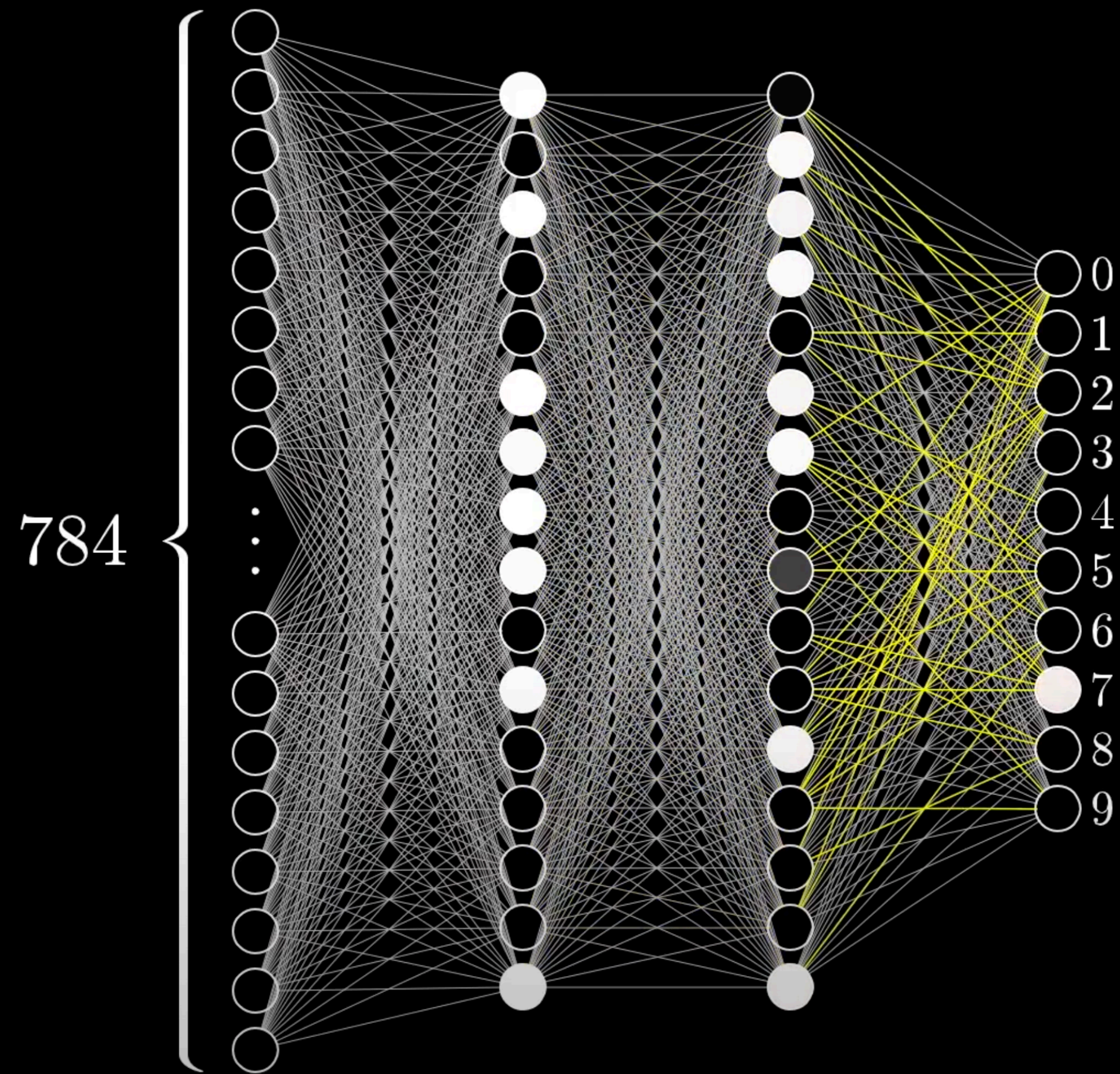


0.97 9

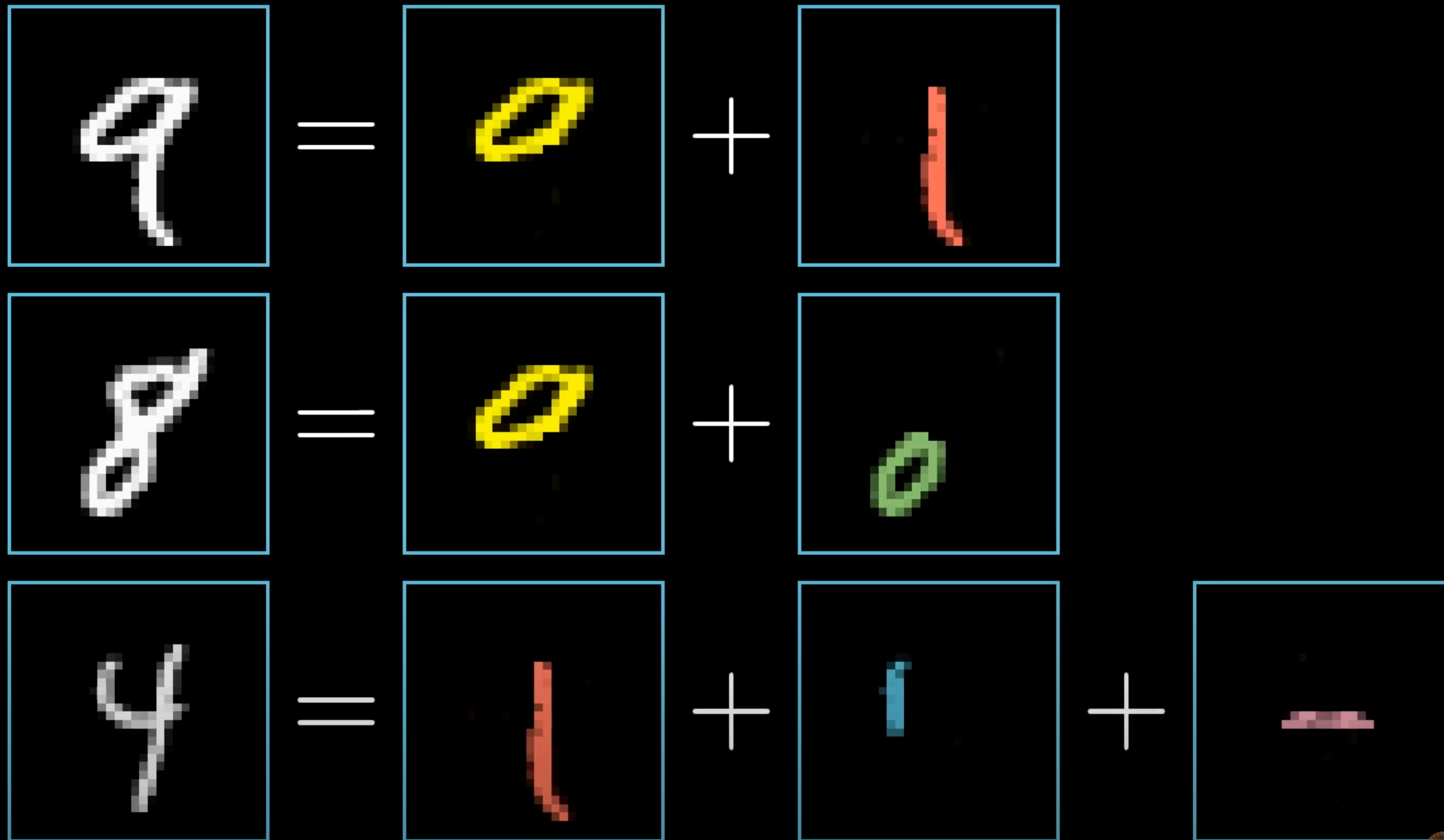
Deep Learning: 101



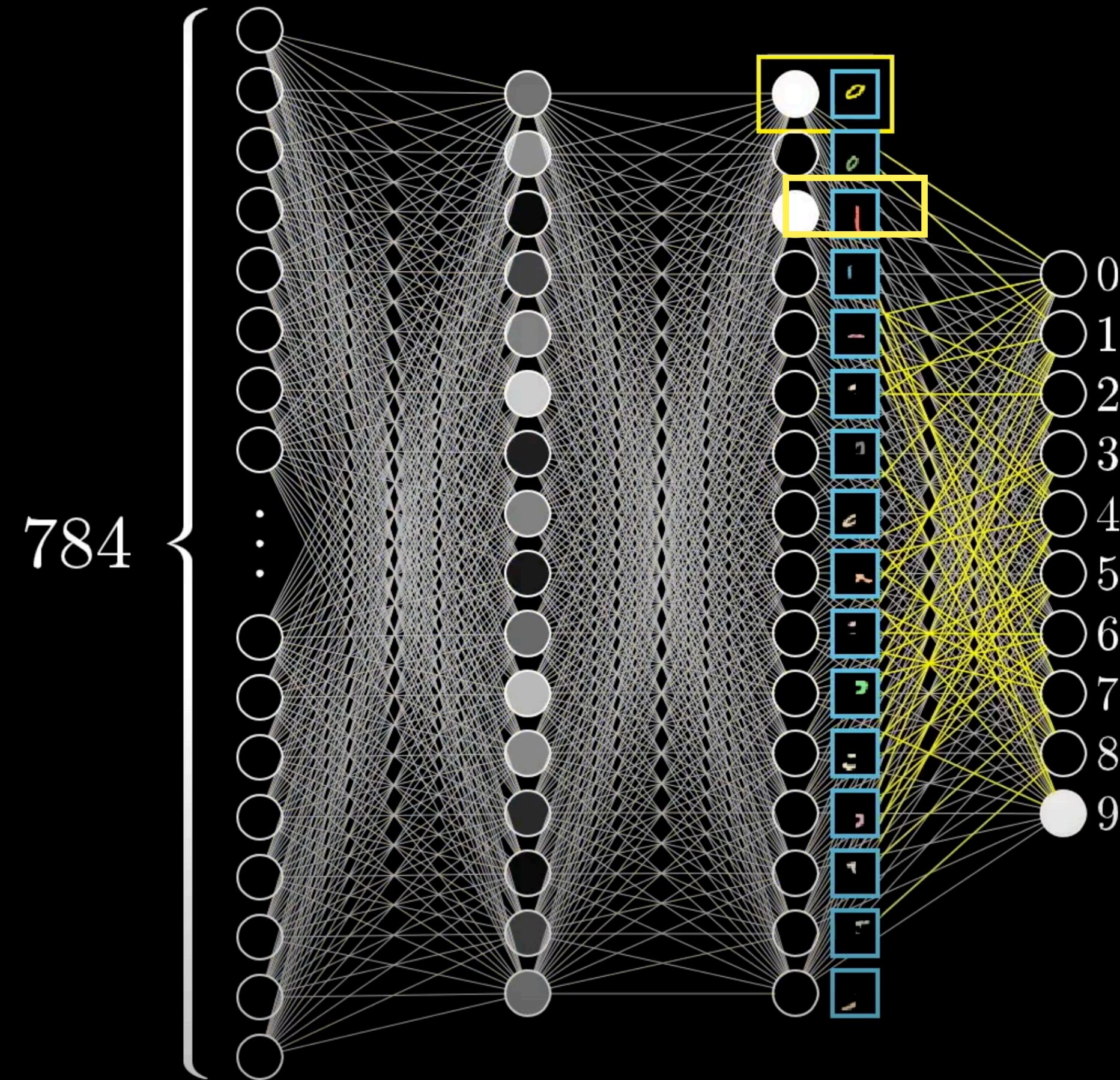
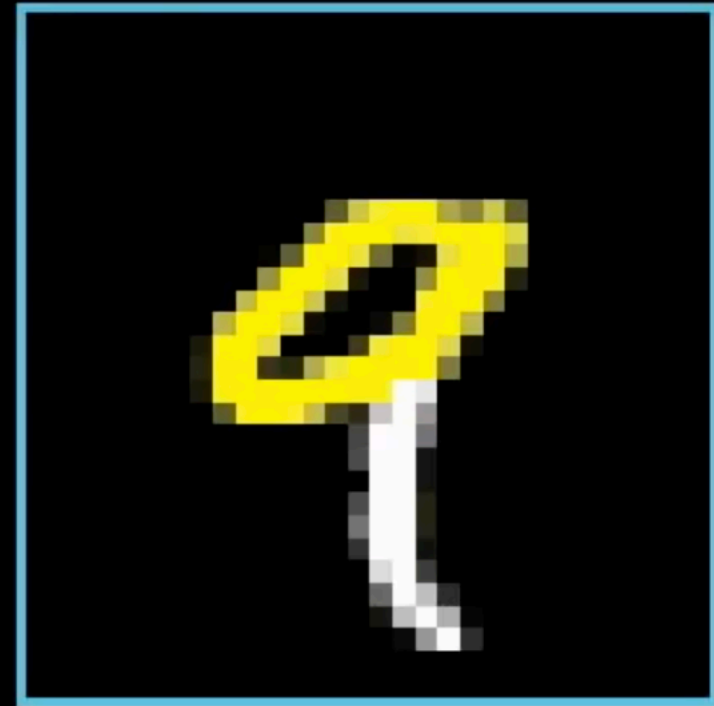
Deep Learning: 101



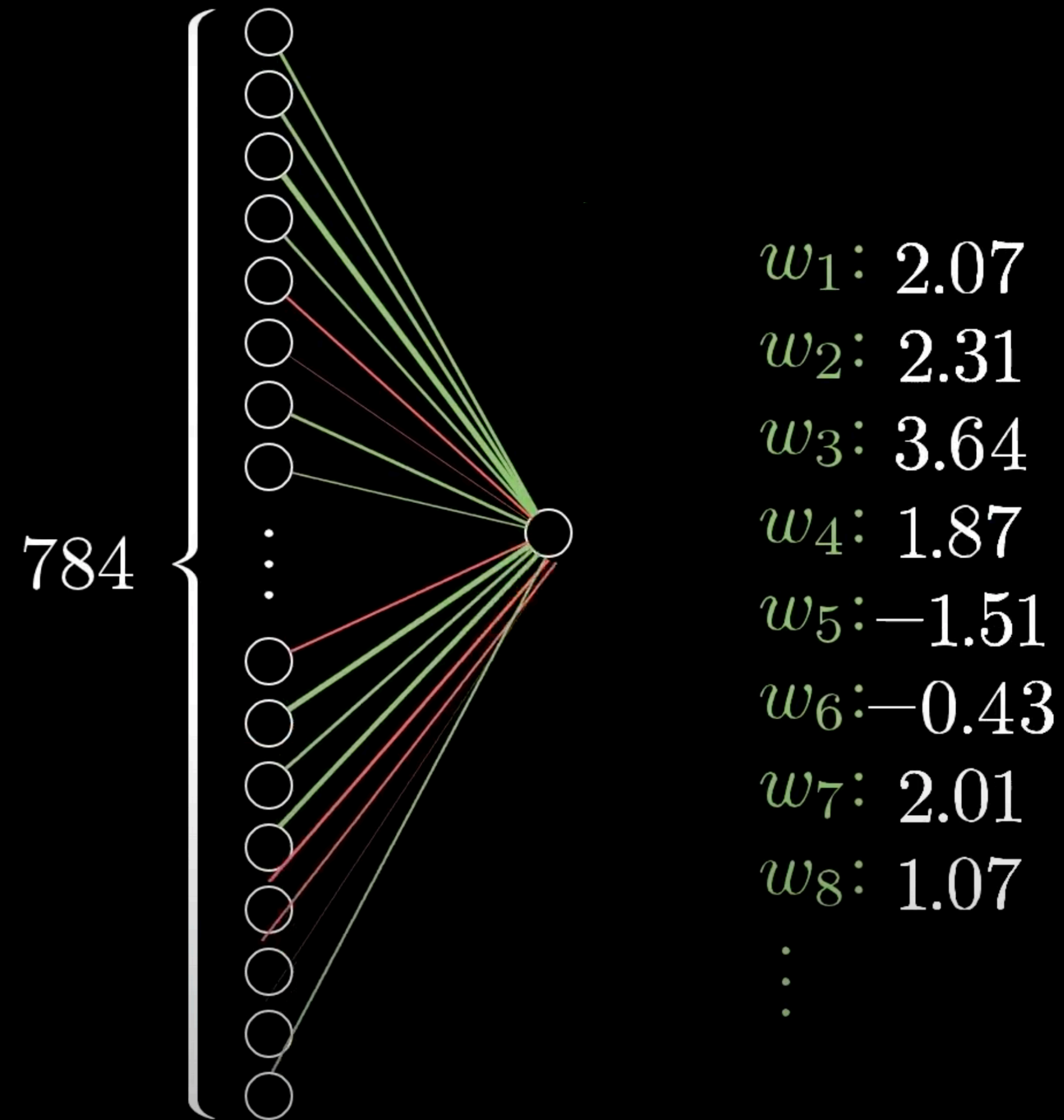
Deep Learning: 101



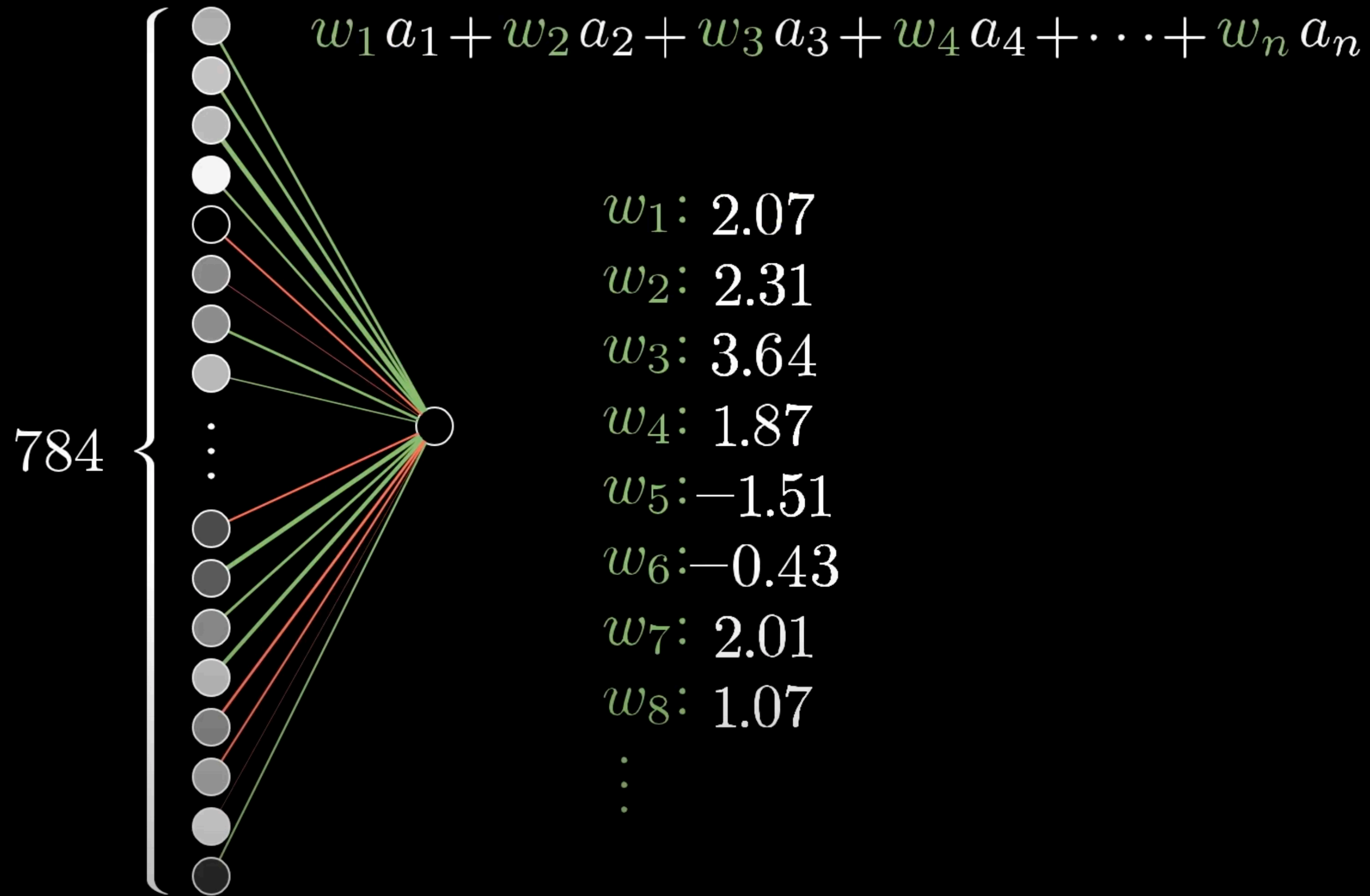
Deep Learning: 101



Deep Learning: 101

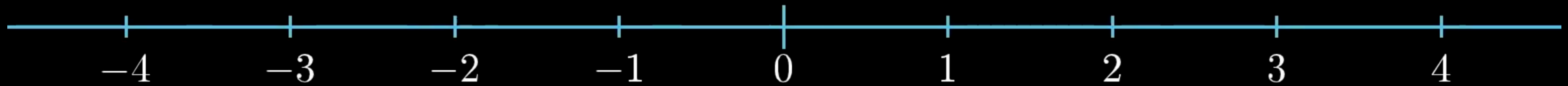


Deep Learning: 101

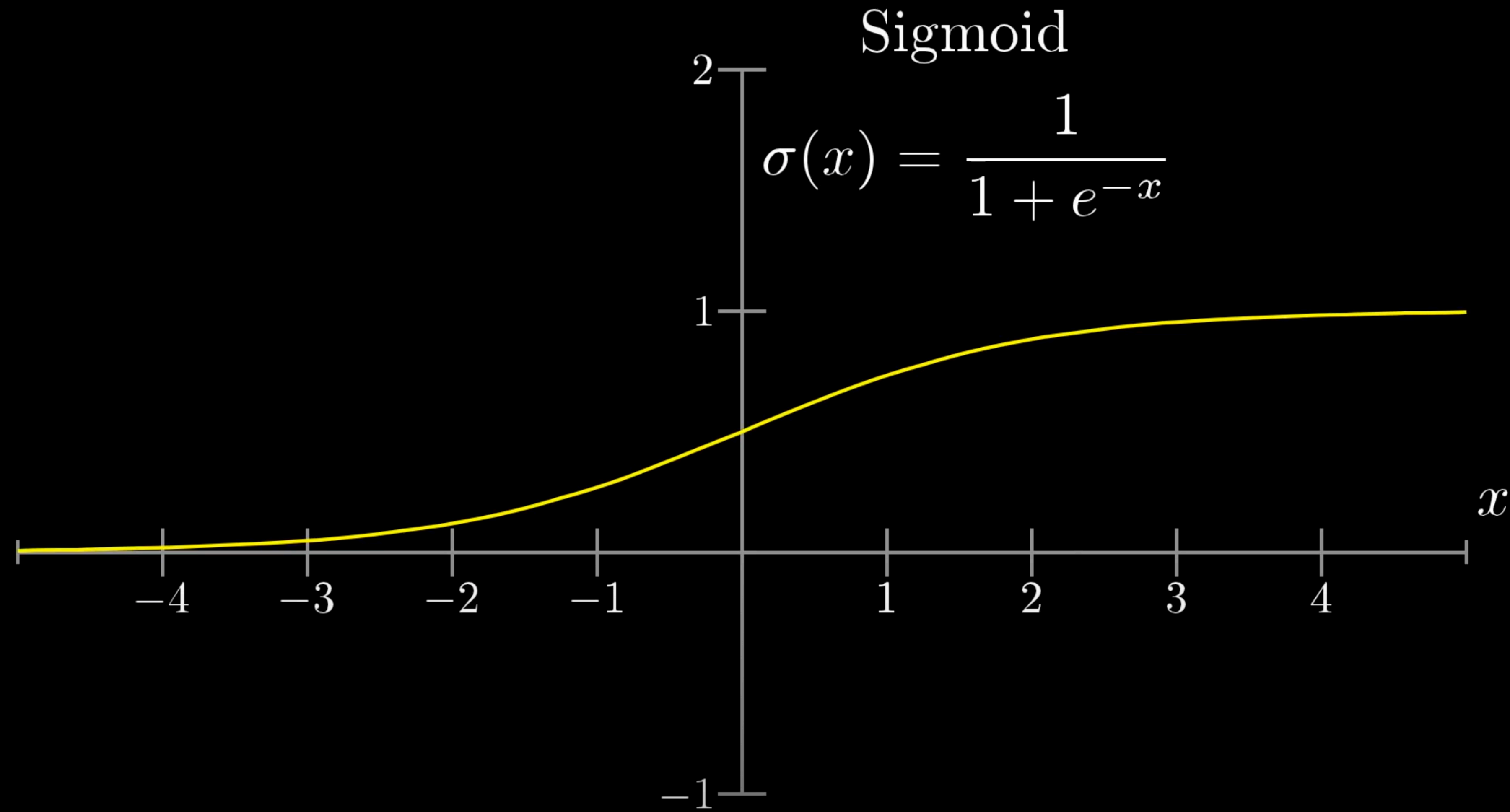


Deep Learning: 101

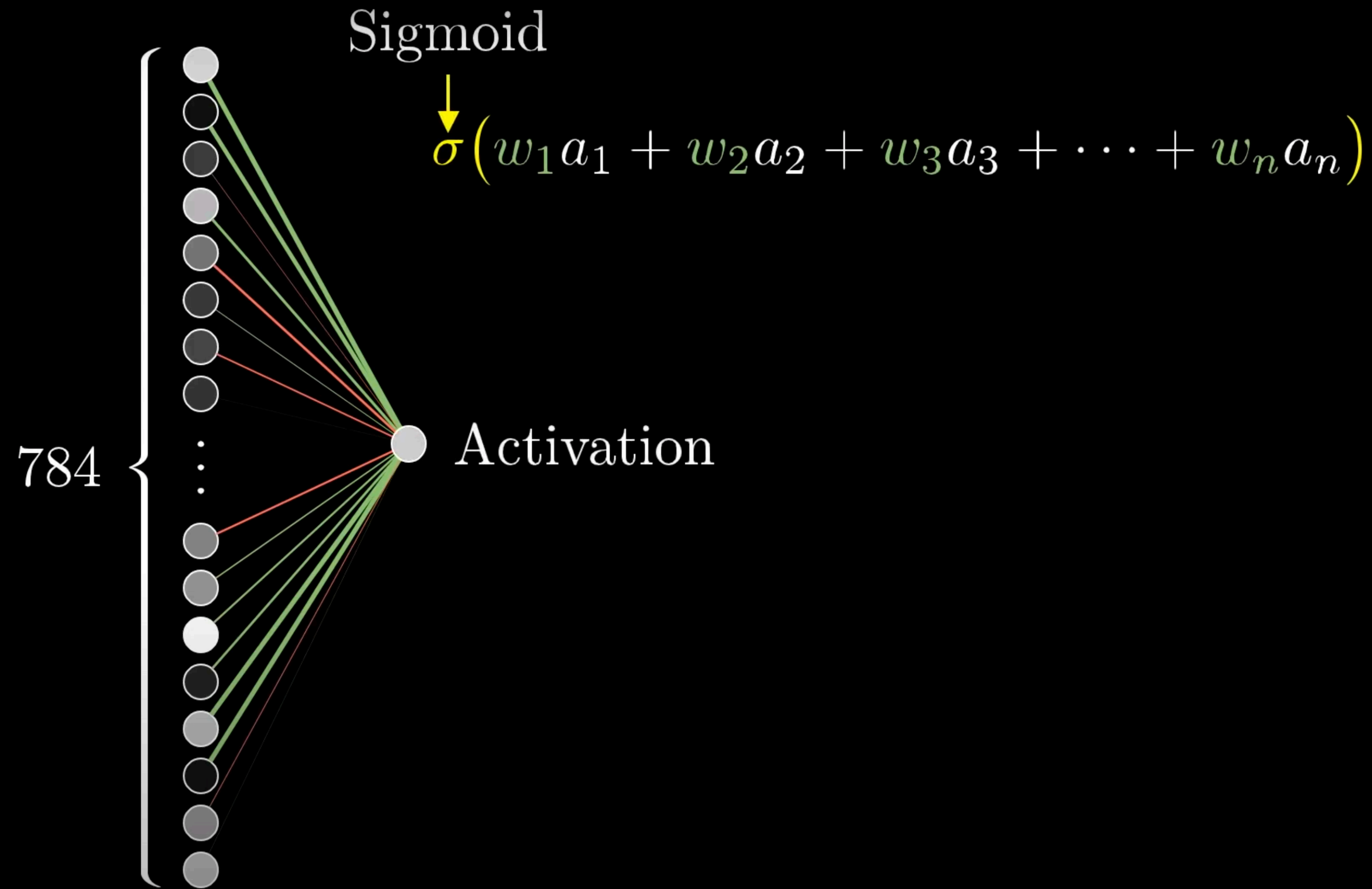
$$w_1 a_1 + w_2 a_2 + w_3 a_3 + w_4 a_4 + \dots + w_n a_n$$



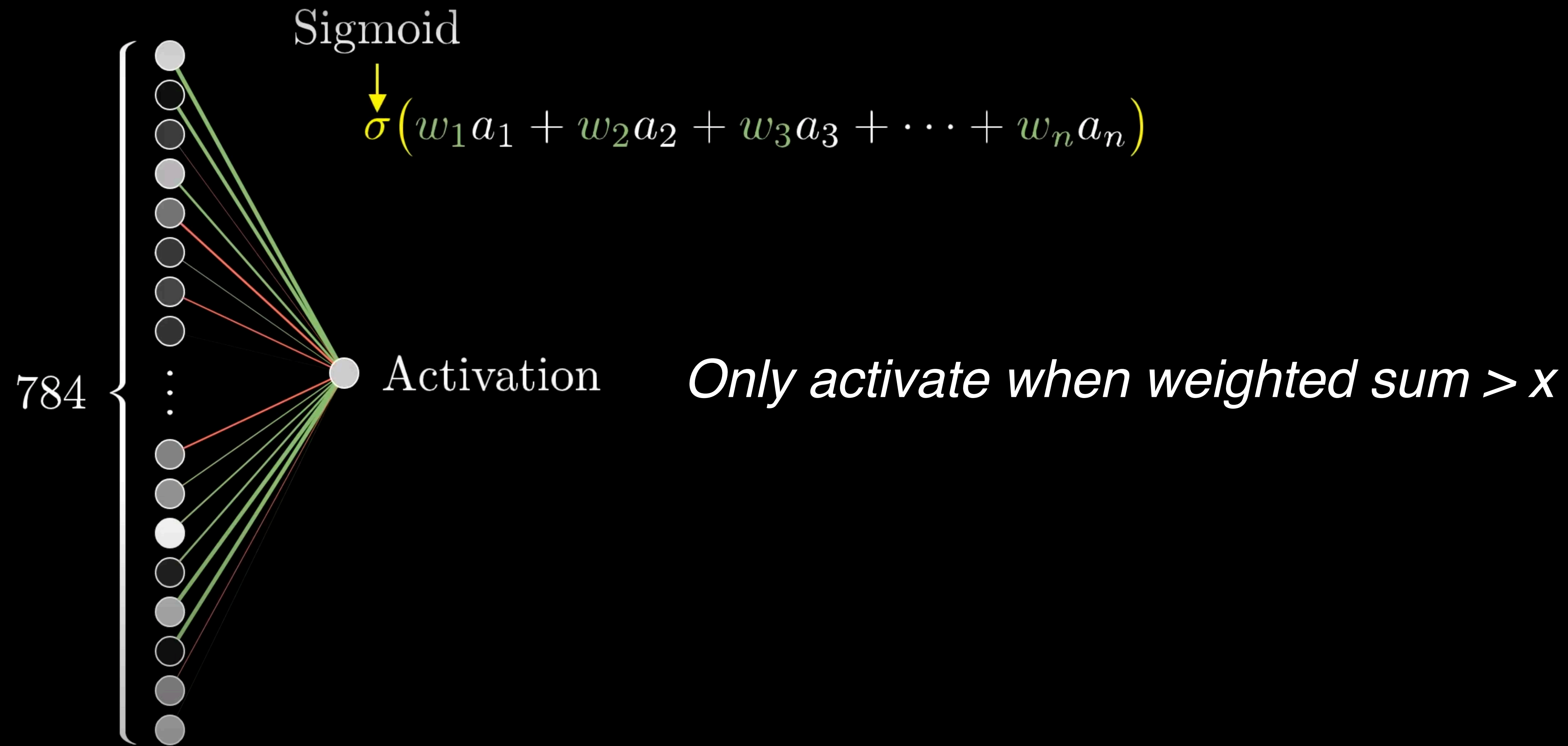
Deep Learning: 101



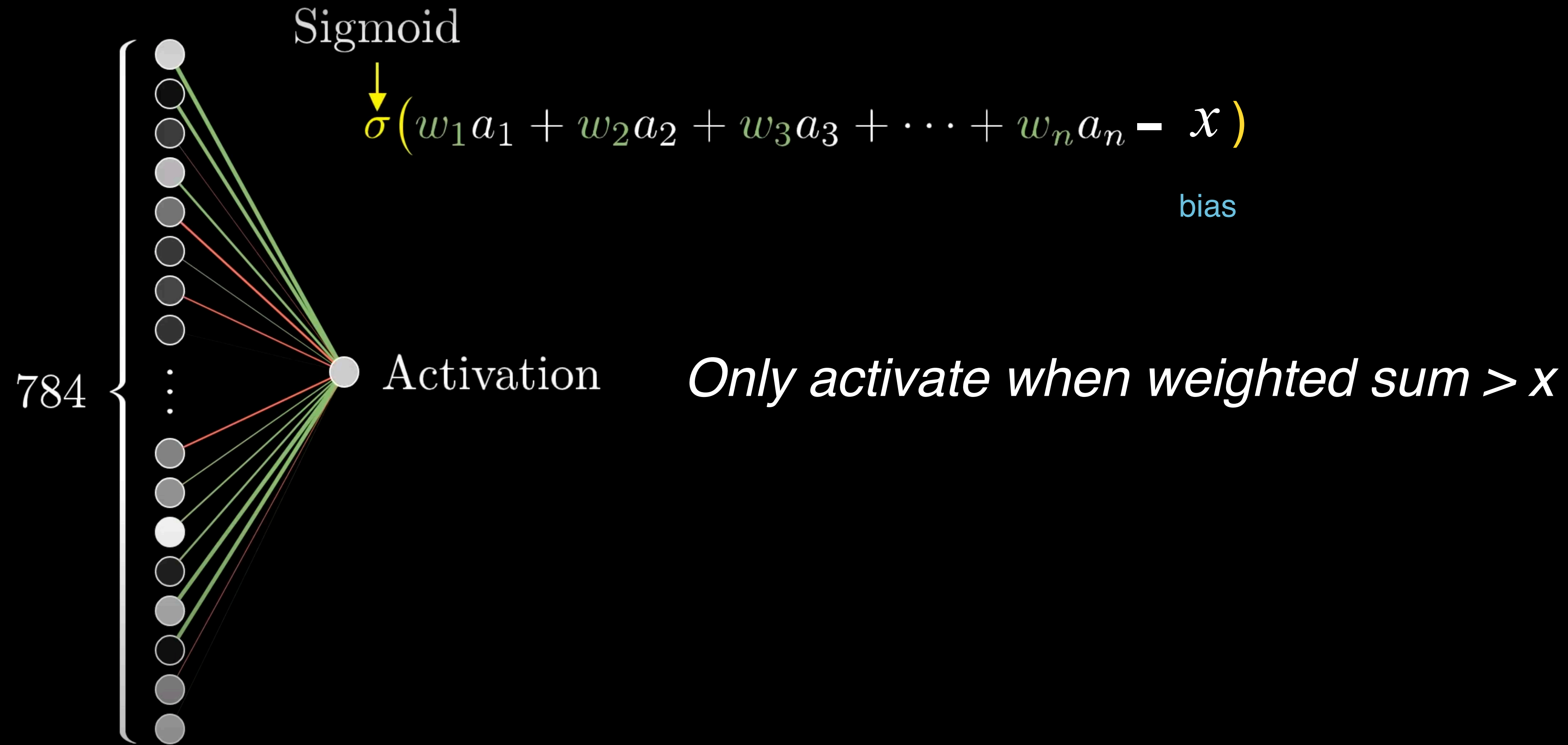
Deep Learning: 101



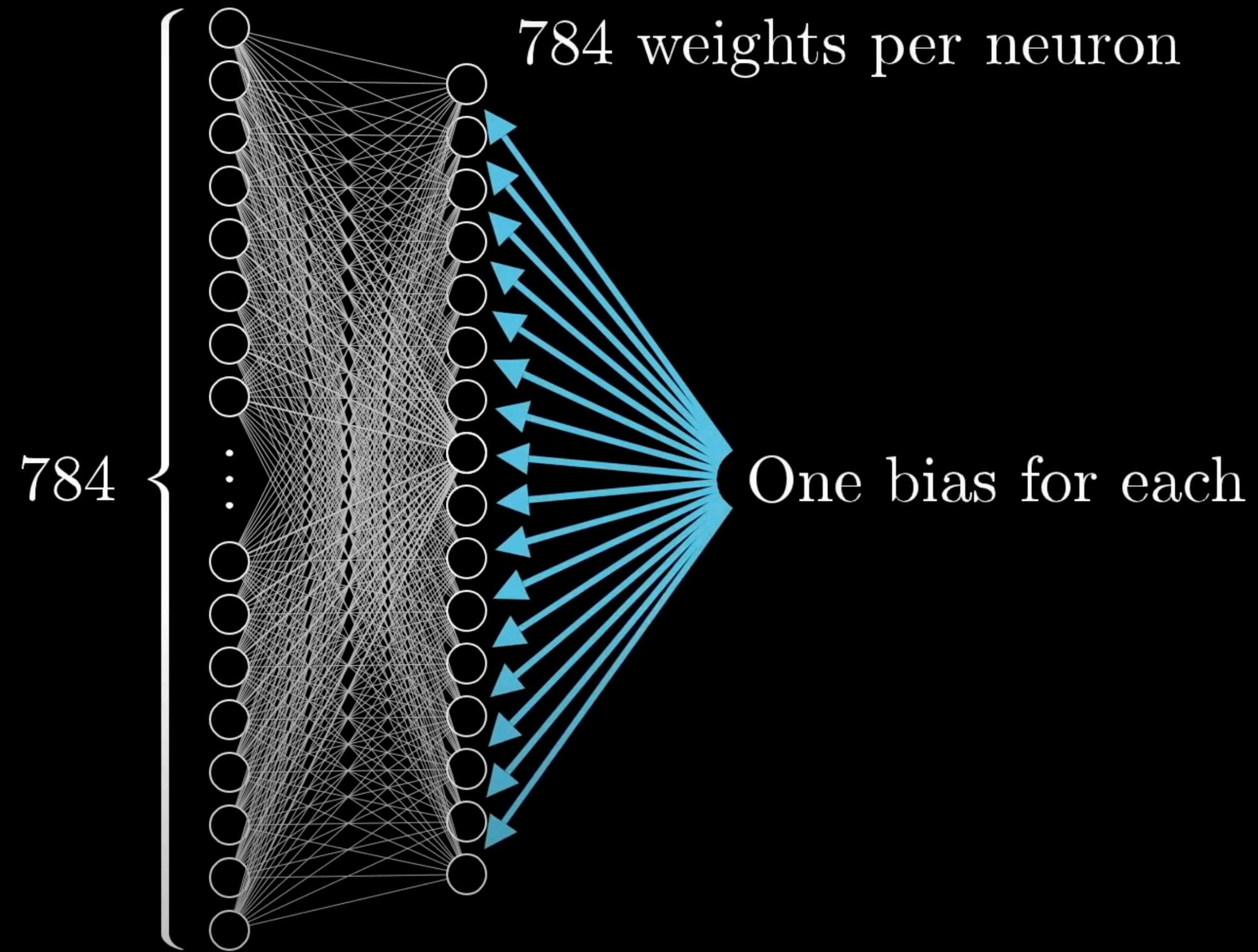
Deep Learning: 101



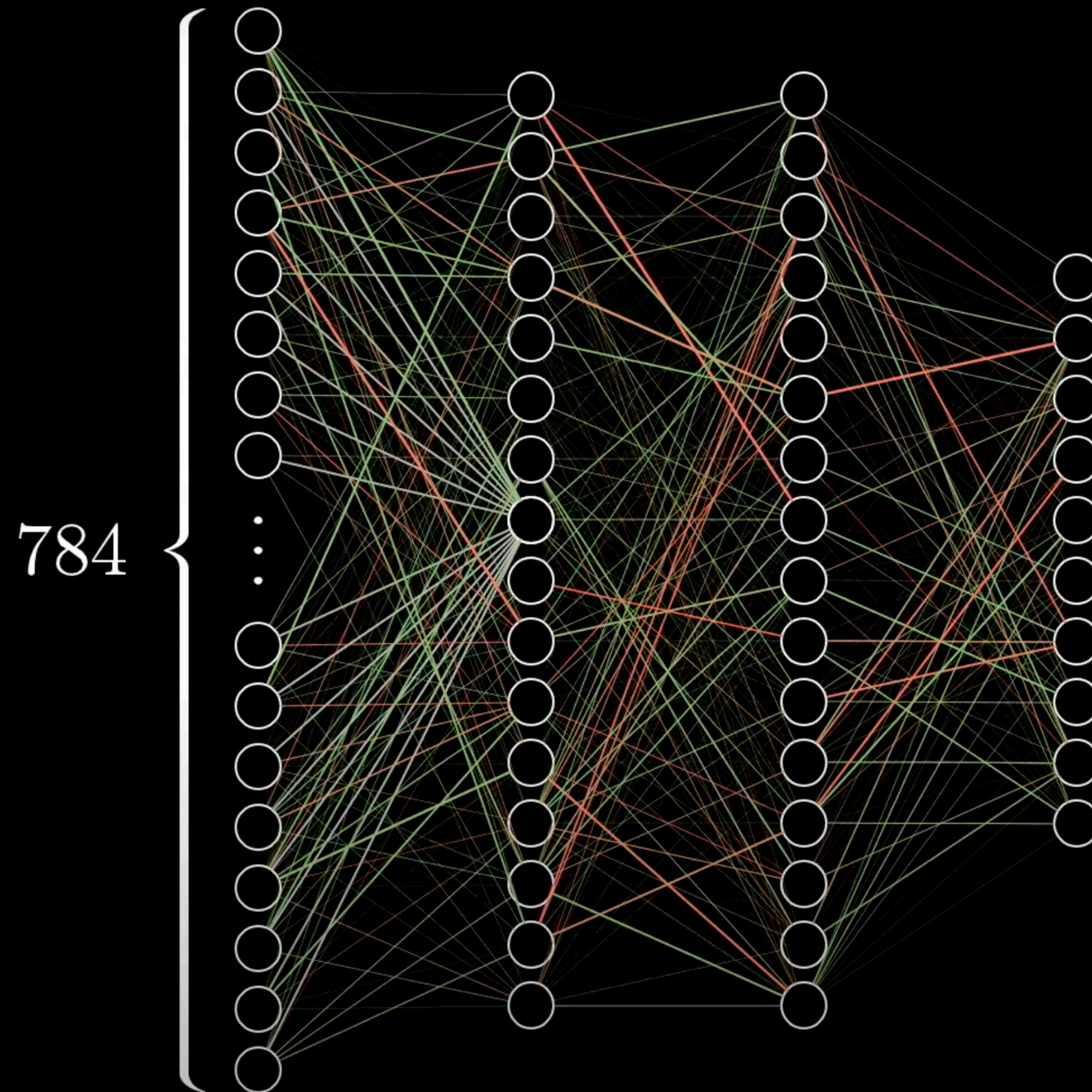
Deep Learning: 101



Deep Learning: 101



Deep Learning: 101



$$784 \times 16 + 16 \times 16 + 16 \times 10$$

weights

$$16 + 16 + 10$$

biases

$$13,002$$



Deep Learning: 101

Learning: Finding the right weights and biases



antoniomastropaolo.com



[aura-se-lab.github.io](https://github.com/aura-se-lab)



Deep Learning: 101

$$\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \dots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} - \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \end{bmatrix}$$

weights activations biases

Learning: Finding the right weights and biases

Deep Learning: 101

Learning: Finding the right weights and biases

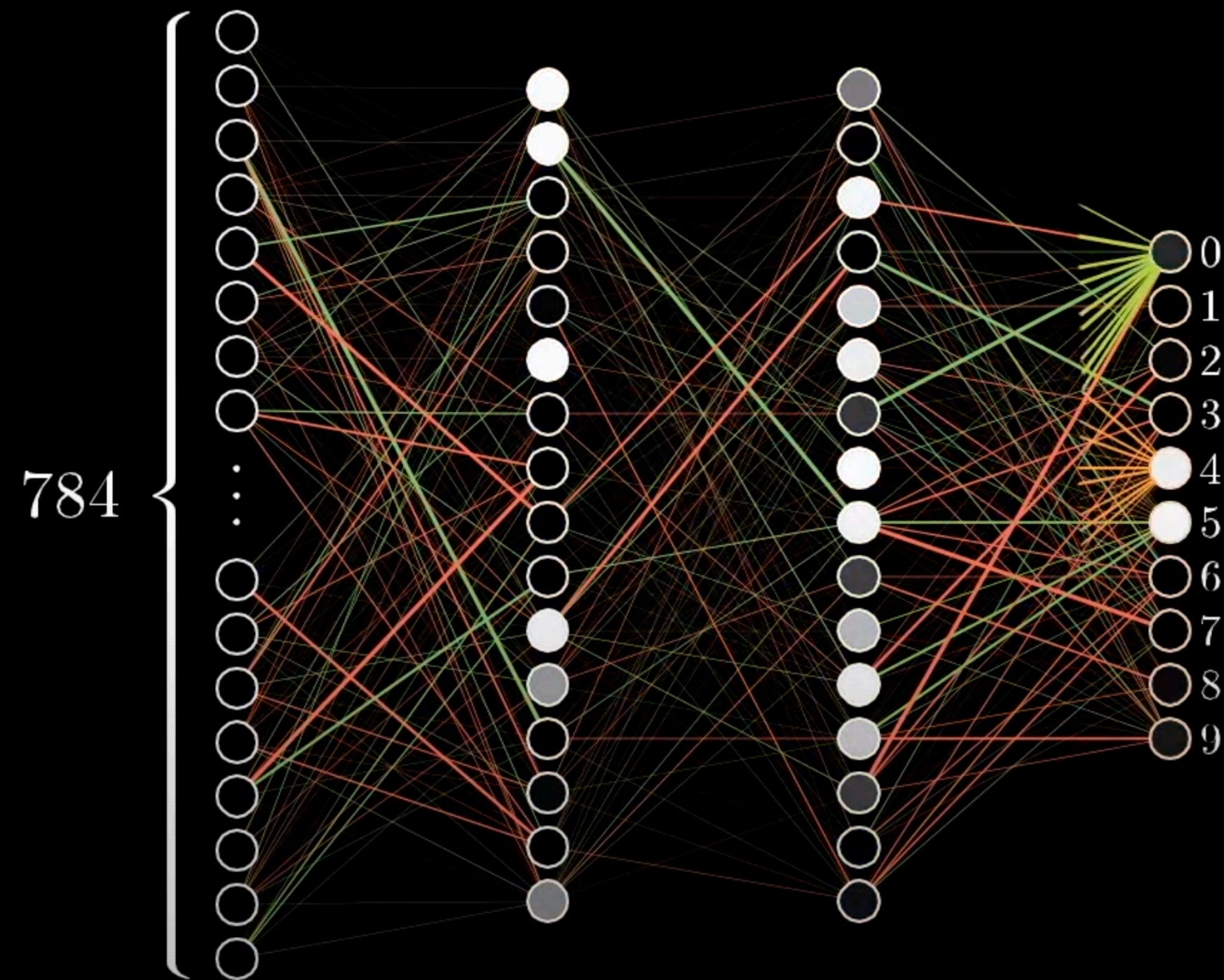
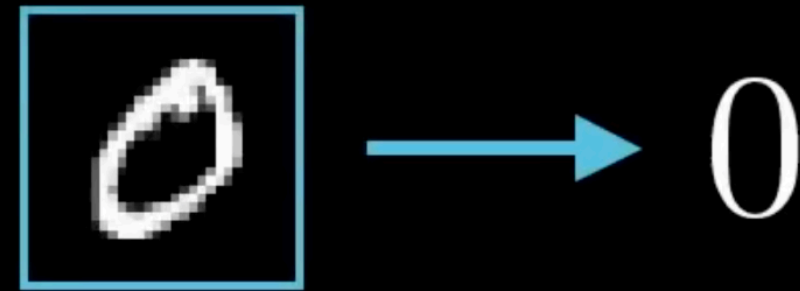
$$\sigma \left(\begin{array}{c} \left[\begin{array}{cccc} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \dots & w_{k,n} \end{array} \right] \left[\begin{array}{c} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{array} \right] - \left[\begin{array}{c} b_0 \\ b_1 \\ \vdots \\ b_k \end{array} \right] \end{array} \right)$$

weights activations biases

Deep Learning: 101

The entire network is “just” a function

Deep Learning: 101



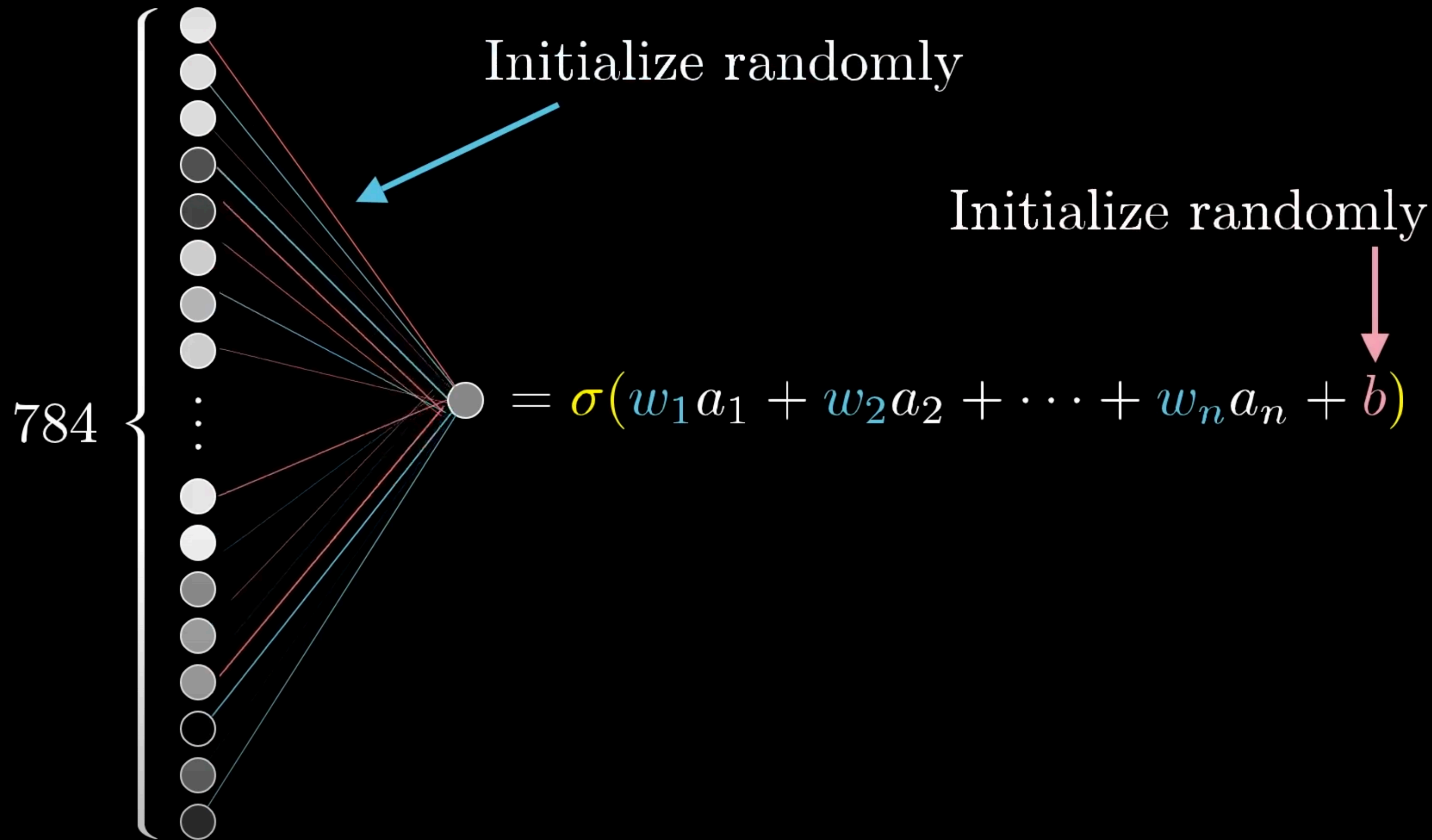
Deep Learning: 101

Labeled data (Training set)

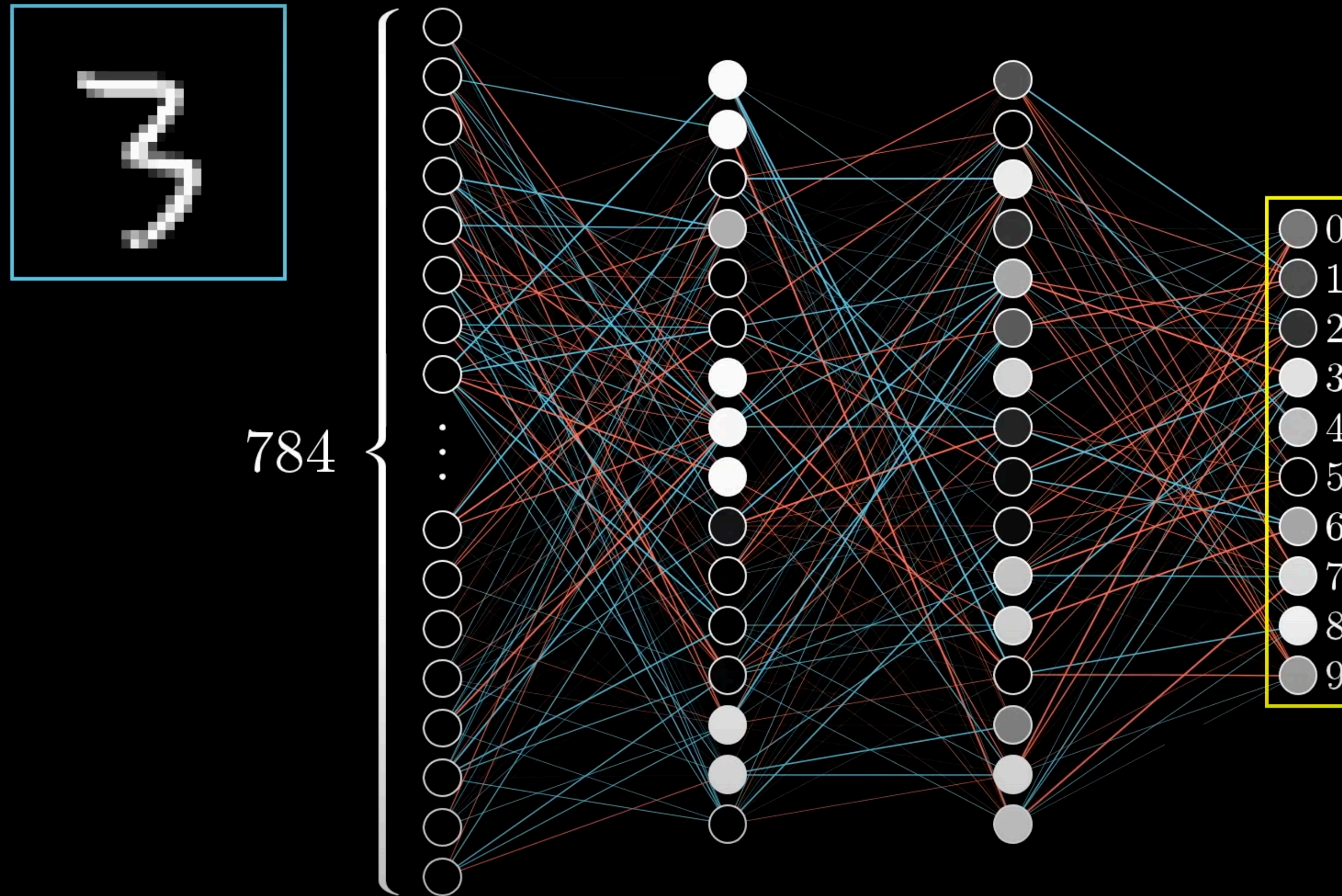
5	→	5
0	→	0
4	→	4
1	→	1
9	→	9
2	→	2
1	→	1
3	→	3
1	→	1
4	→	4



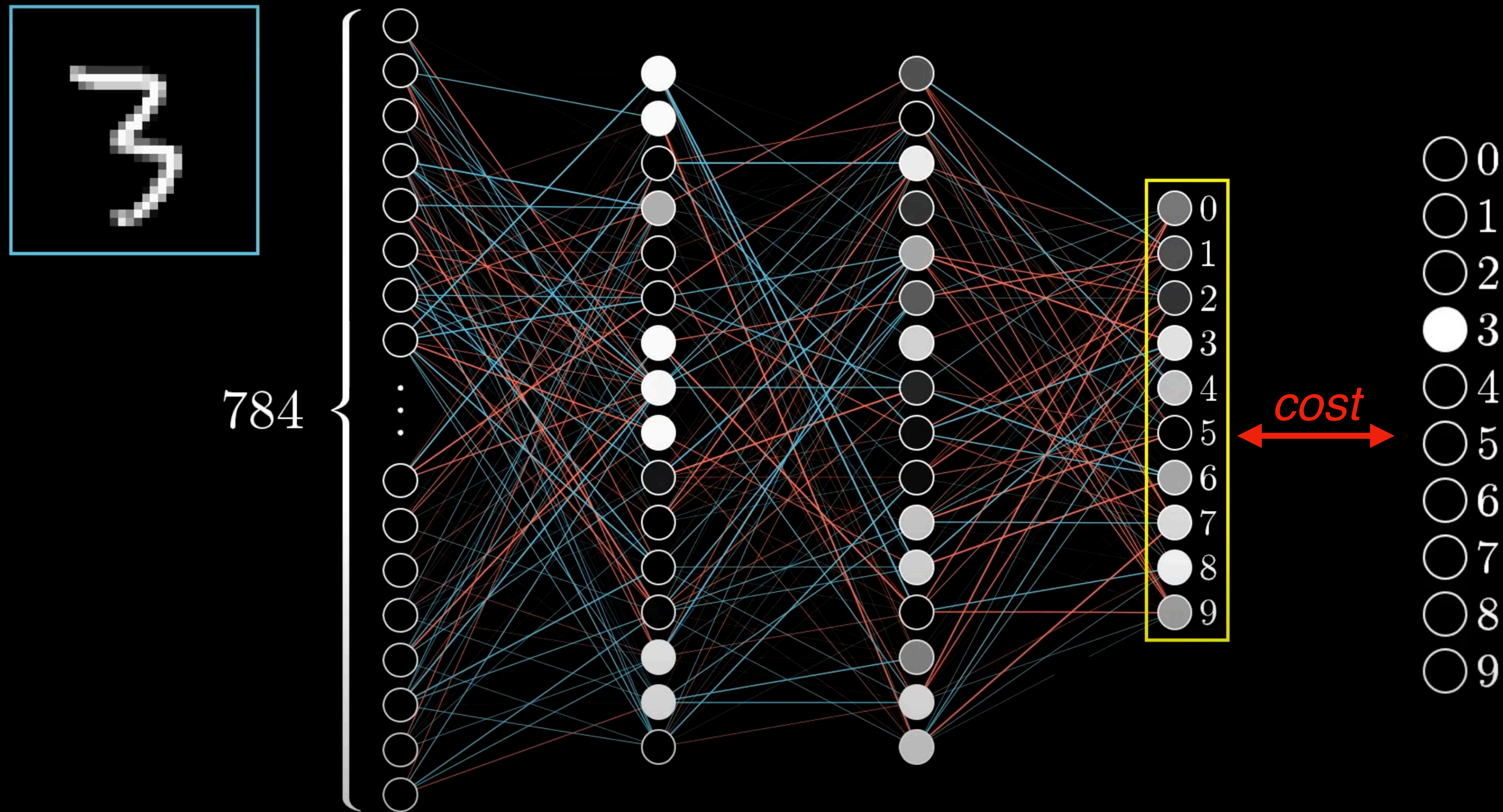
Deep Learning: 101



Deep Learning: 101



Deep Learning: 101

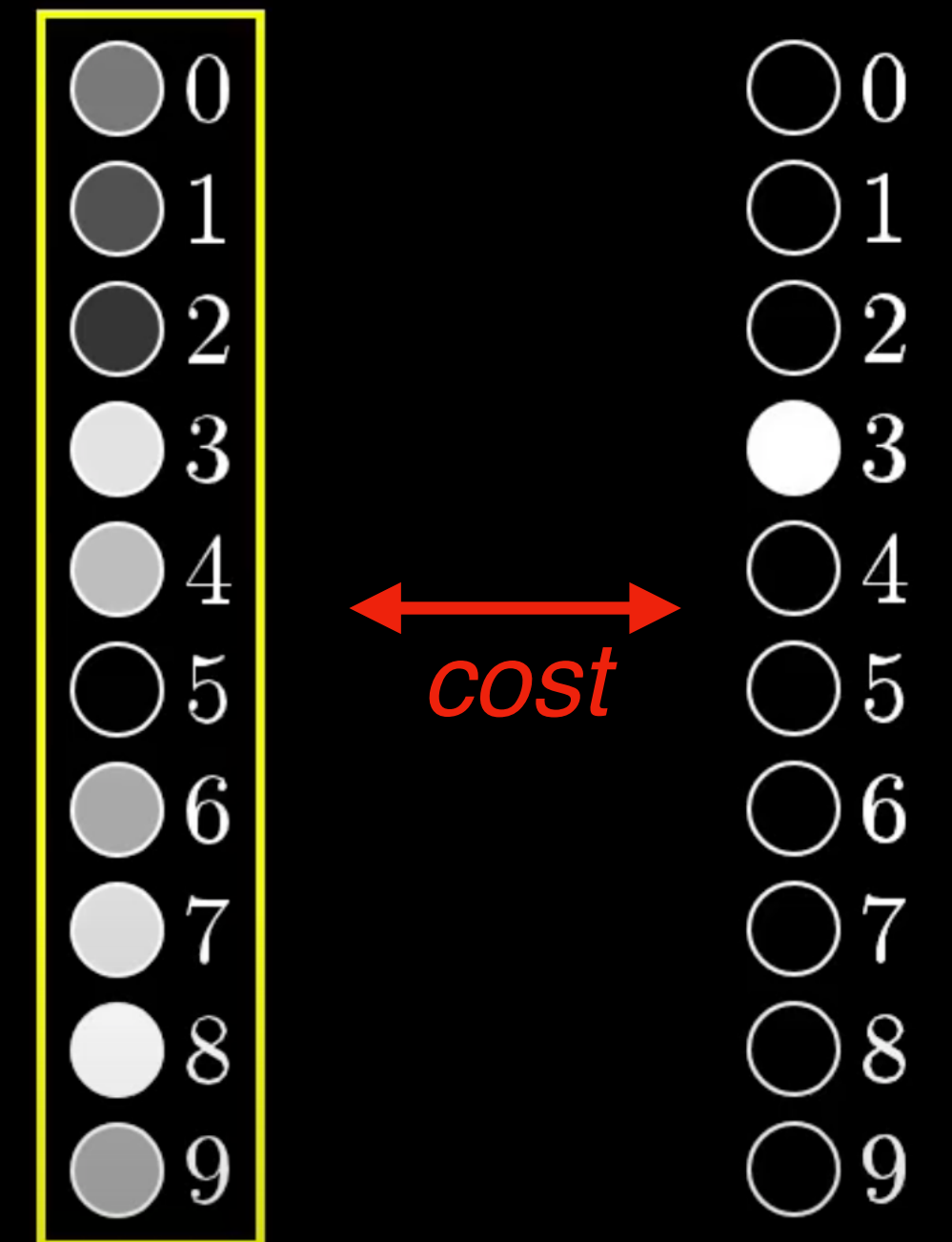


Deep Learning: 101

Cost of **3**

3.37

$$\begin{aligned} 0.1863 &\leftarrow (0.43 - 0.00)^2 + \\ 0.0809 &\leftarrow (0.28 - 0.00)^2 + \\ 0.0357 &\leftarrow (0.19 - 0.00)^2 + \\ 0.0138 &\leftarrow (0.88 - 1.00)^2 + \\ 0.5242 &\leftarrow (0.72 - 0.00)^2 + \\ 0.0001 &\leftarrow (0.01 - 0.00)^2 + \\ 0.4079 &\leftarrow (0.64 - 0.00)^2 + \\ 0.7388 &\leftarrow (0.86 - 0.00)^2 + \\ 0.9817 &\leftarrow (0.99 - 0.00)^2 + \\ 0.3998 &\leftarrow (0.63 - 0.00)^2 \end{aligned}$$



Deep Learning: 101

*We consider the average **cost** of the predictions on all **training data***



antoniomastropaolo.com



[aura-se-lab.github.io](https://github.com/aura-se-lab)



Deep Learning: 101

Cost function to be minimized

Input: 13,002 weights/biases

Output: 1 number (cost)



Deep Learning: 101

Cost function to be minimized

Input: 13,002 weights/biases

Output: 1 number (cost)

*Considering the complexity of this function,
we look for **local minimum**, not global minimum*



Deep Learning: 101

*This is done efficiently using the **backpropagation** algorithm*

