

INTERNATIONAL WORKSHOP ON ADVANCED DEEP LEARNING (IWADL) AT IIT MANDI

Part B: Fully Connected Layers and Backprop

Aditya Nigam, Assistant Professor

School of Computing and Electrical Engineering (SCEE)

Indian Institute of Technology, Mandi

<http://faculty.iitmandi.ac.in/~aditya/>

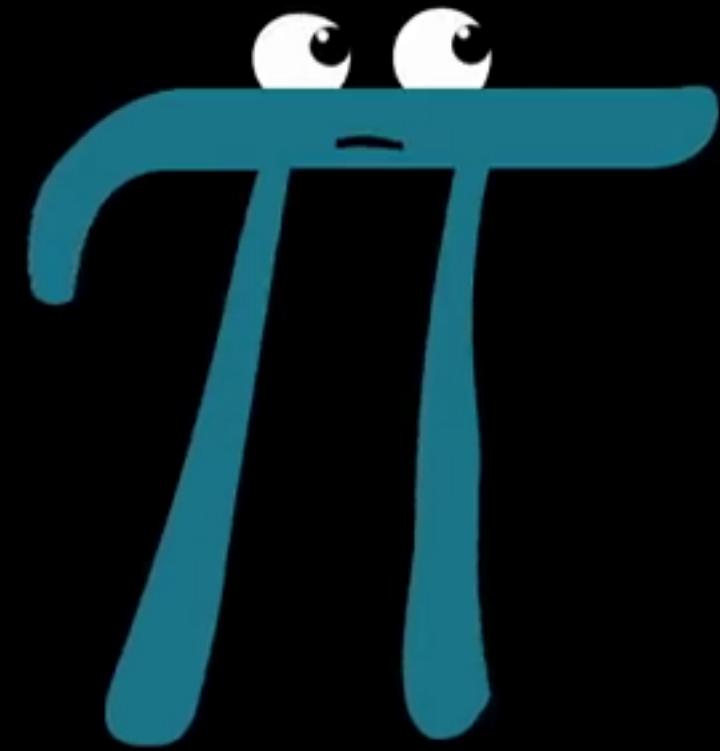
Email : aditya@iitmandi.ac.in



Presentation at IWADL@IIT Mandi

(*Slides : But what *is* a Neural Network? — Chapter 1, deep learning by 3Blue1Brown)

<https://www.youtube.com/watch?v=aircAruvnKk&index=6&list=PL7glvrW4cSVNjYKRxS7okEEhuZs8UpvpE&t=964s>

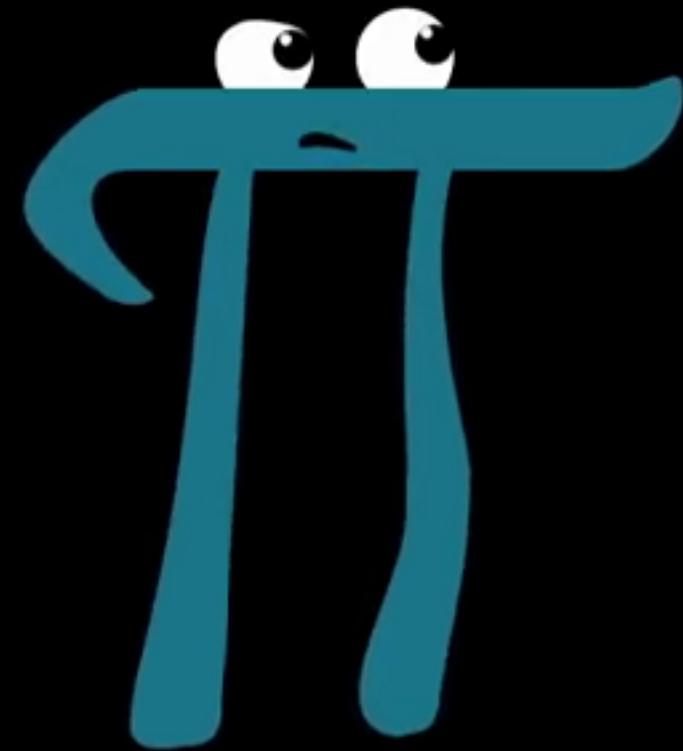


28px

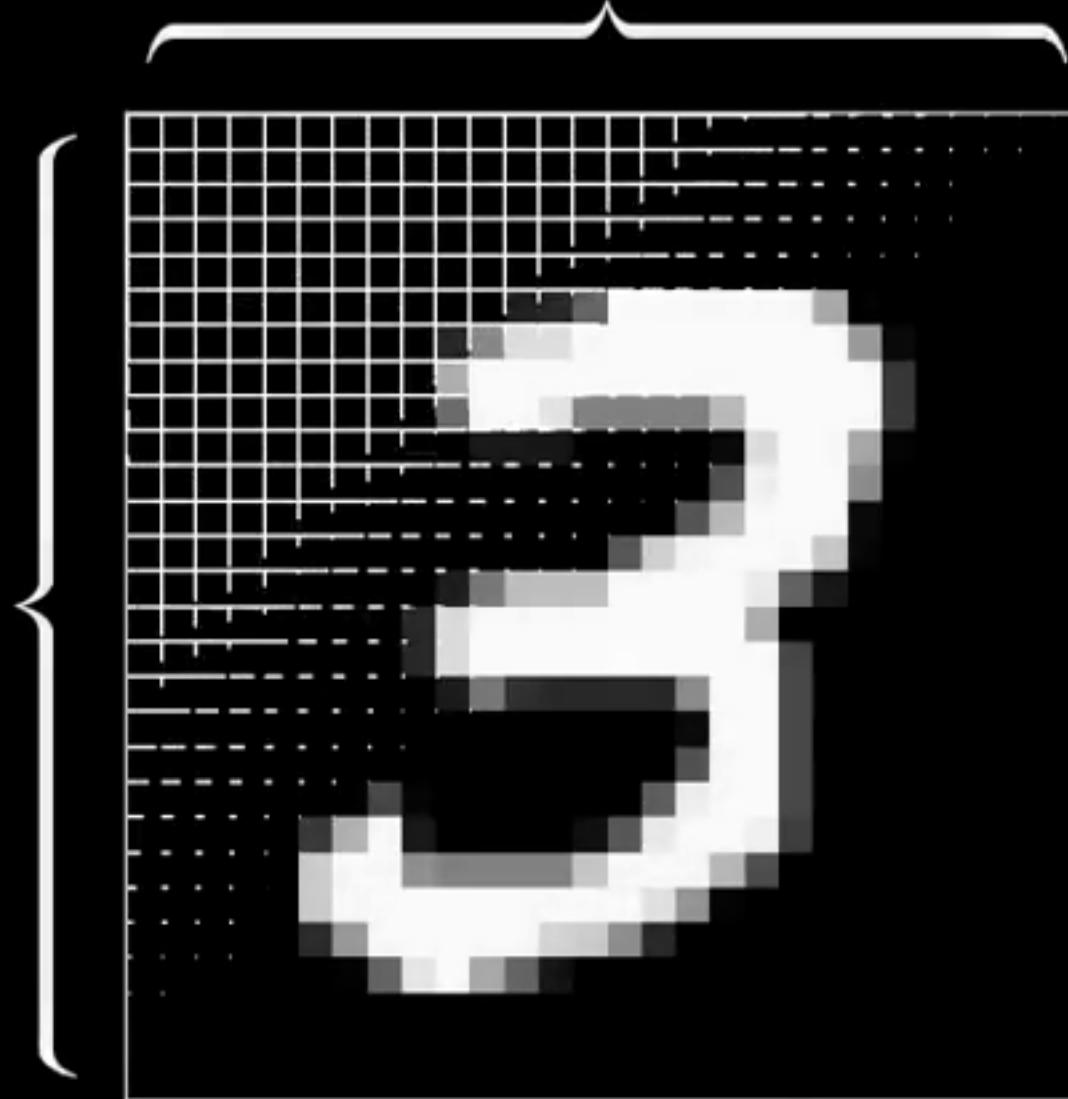
{



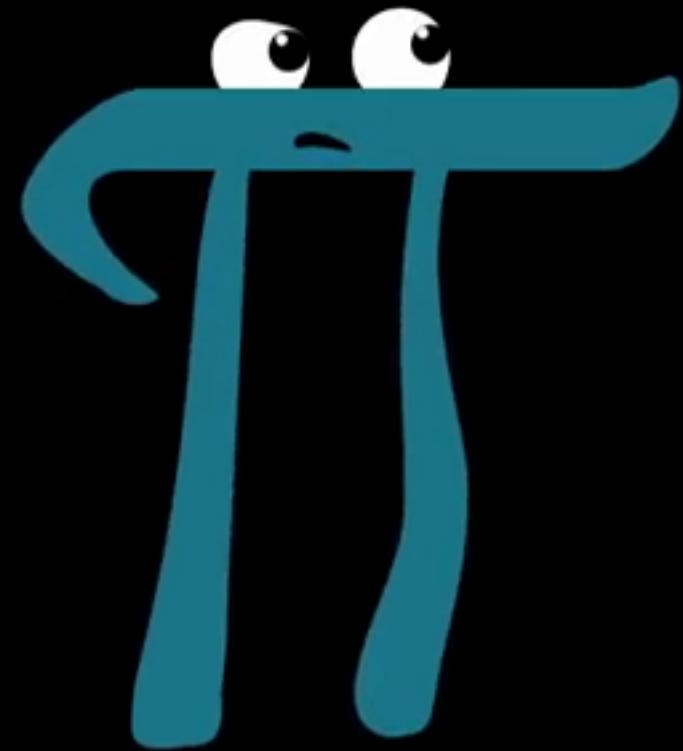
28px



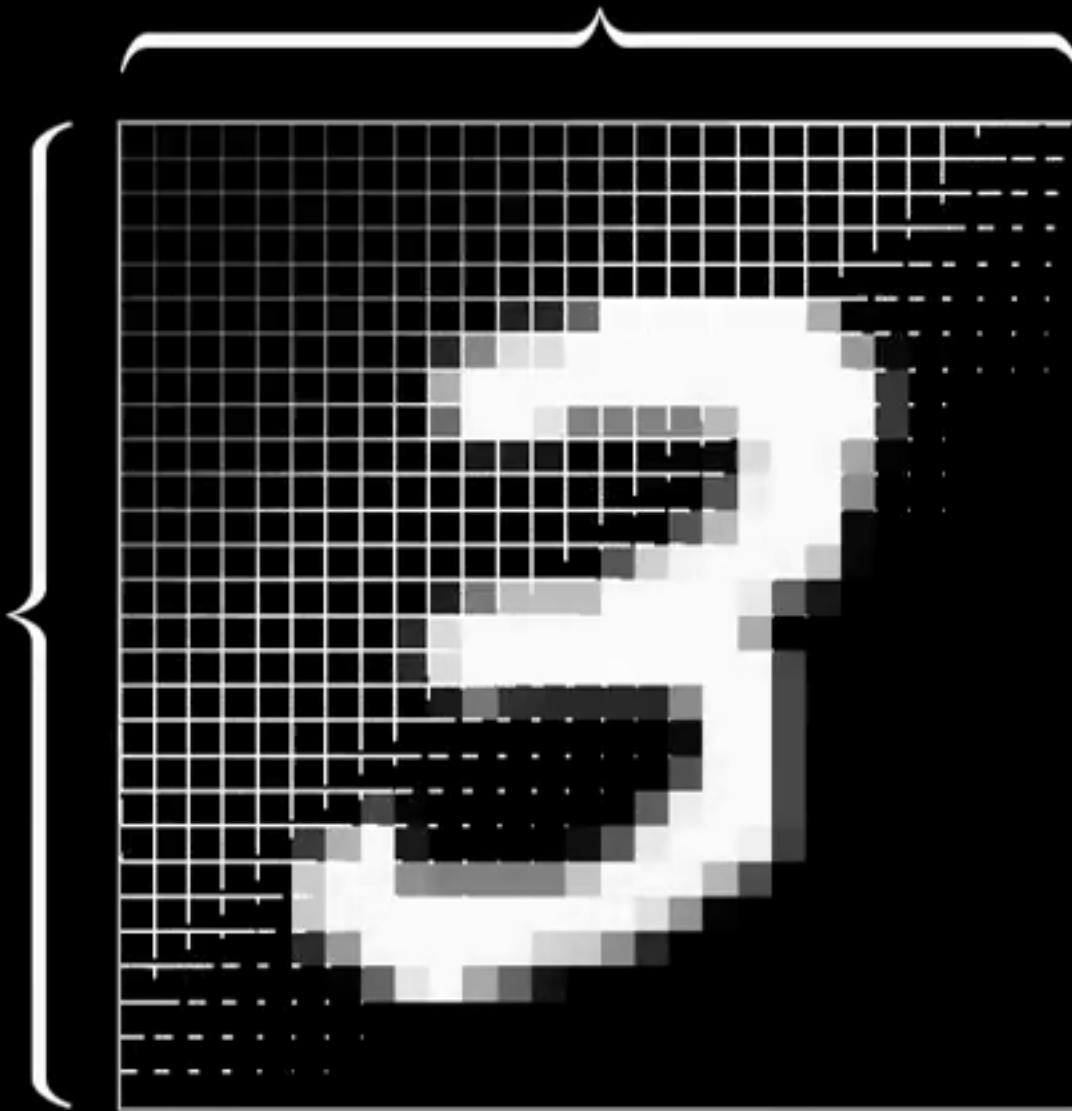
28px



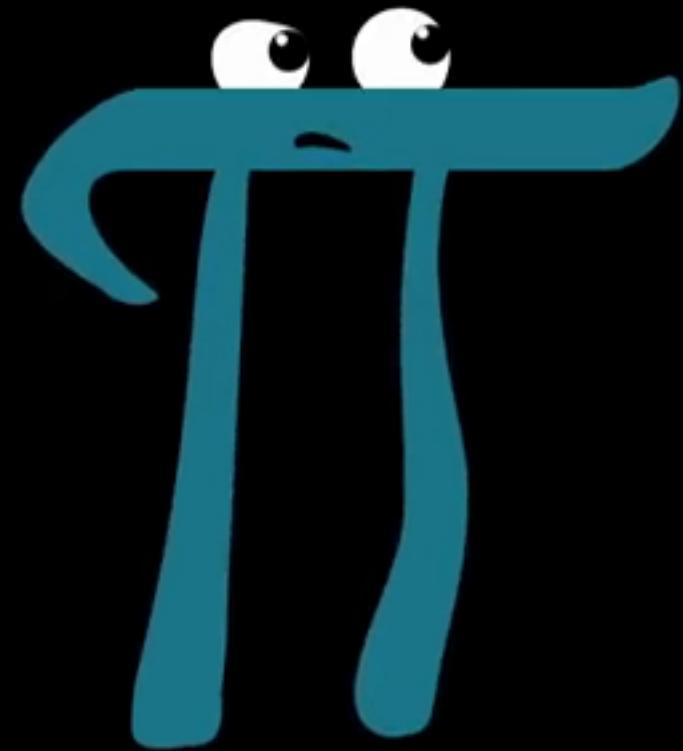
28px



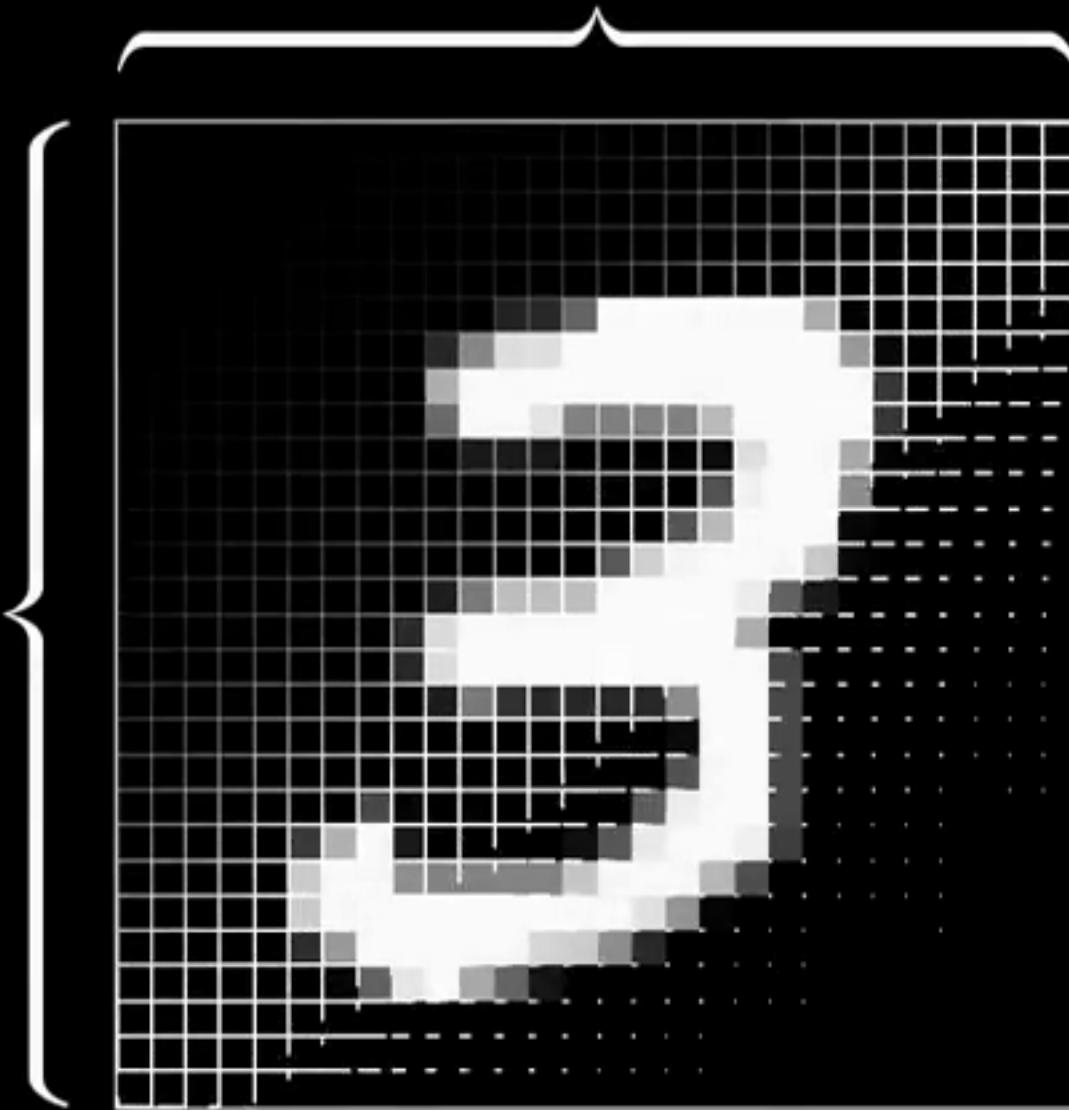
28px



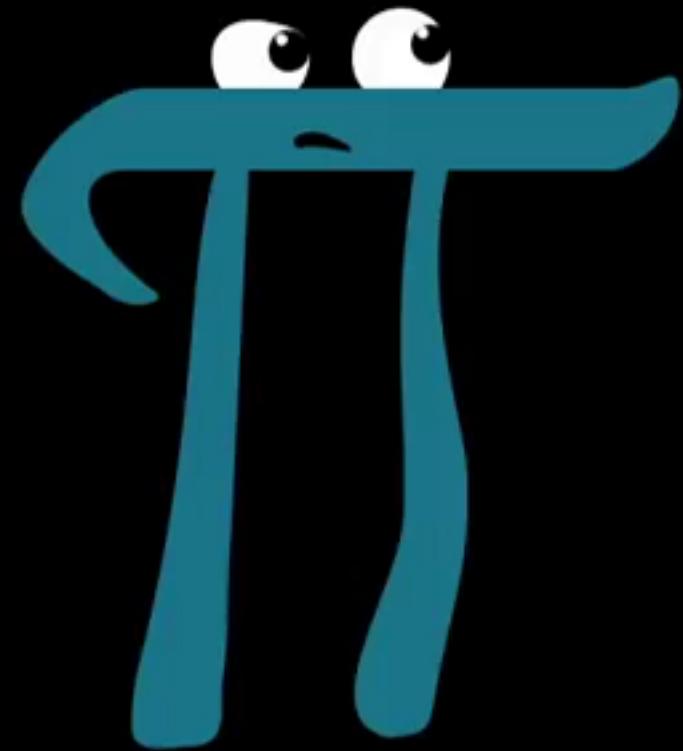
28px



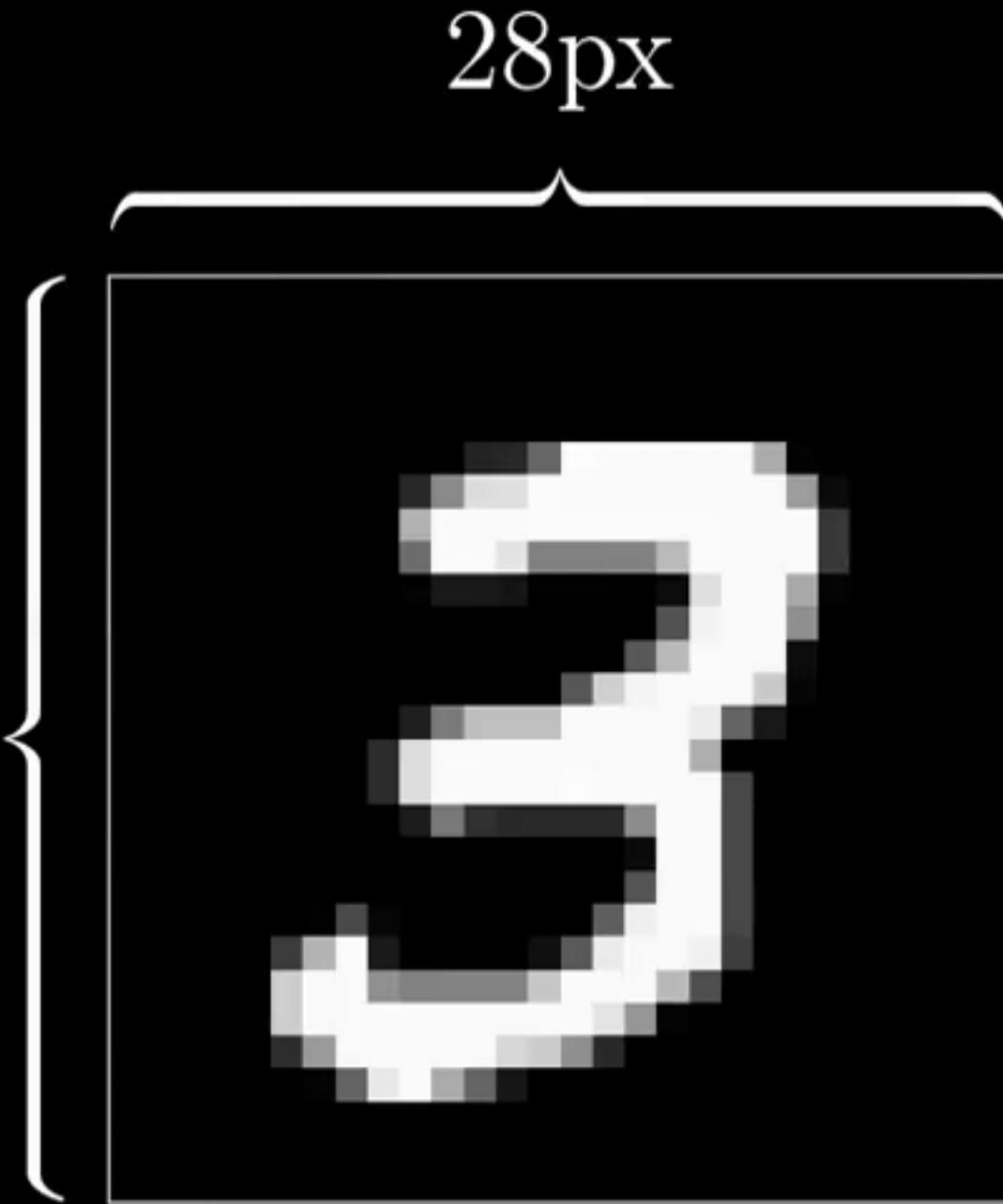
28px



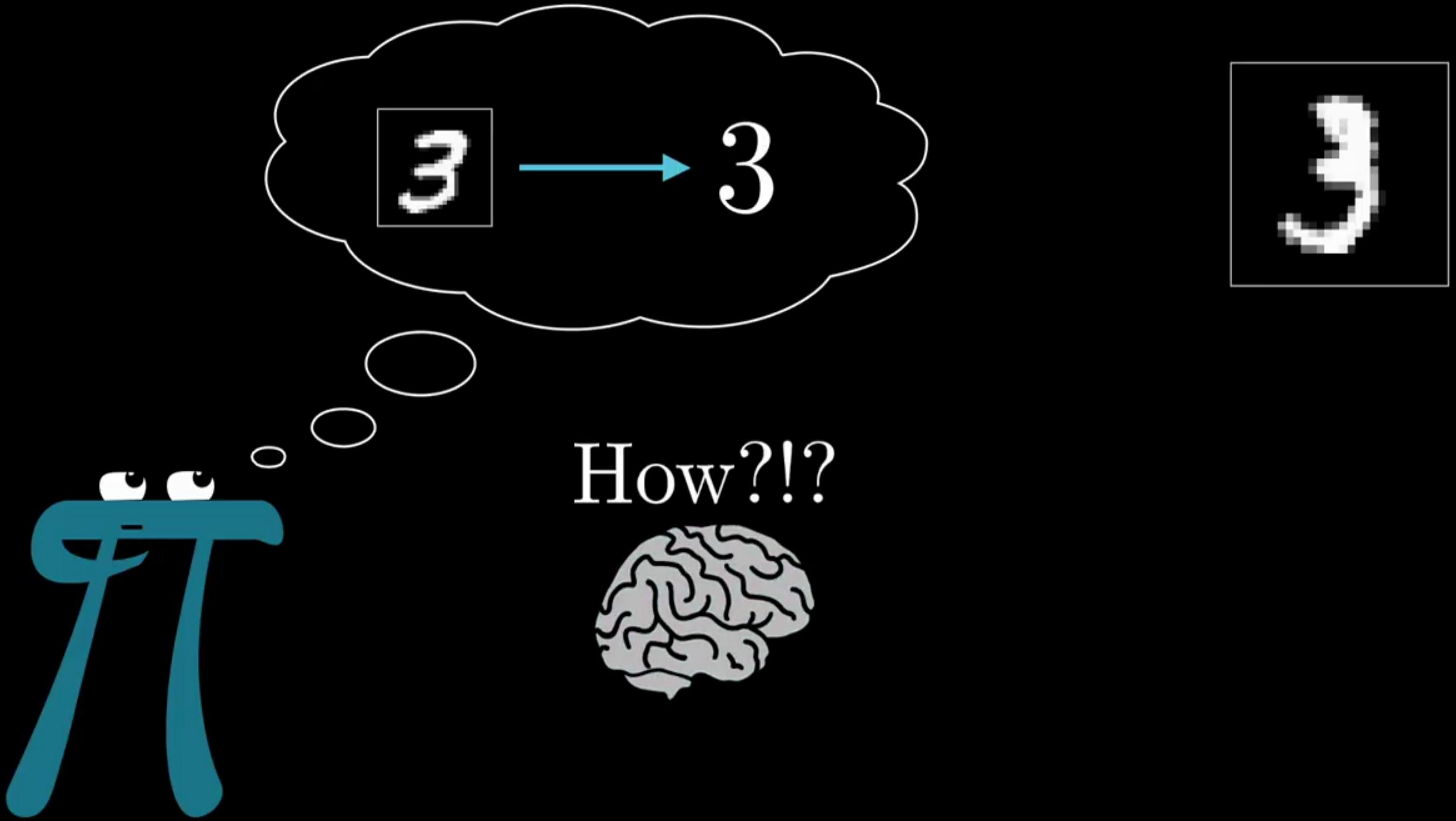
28px

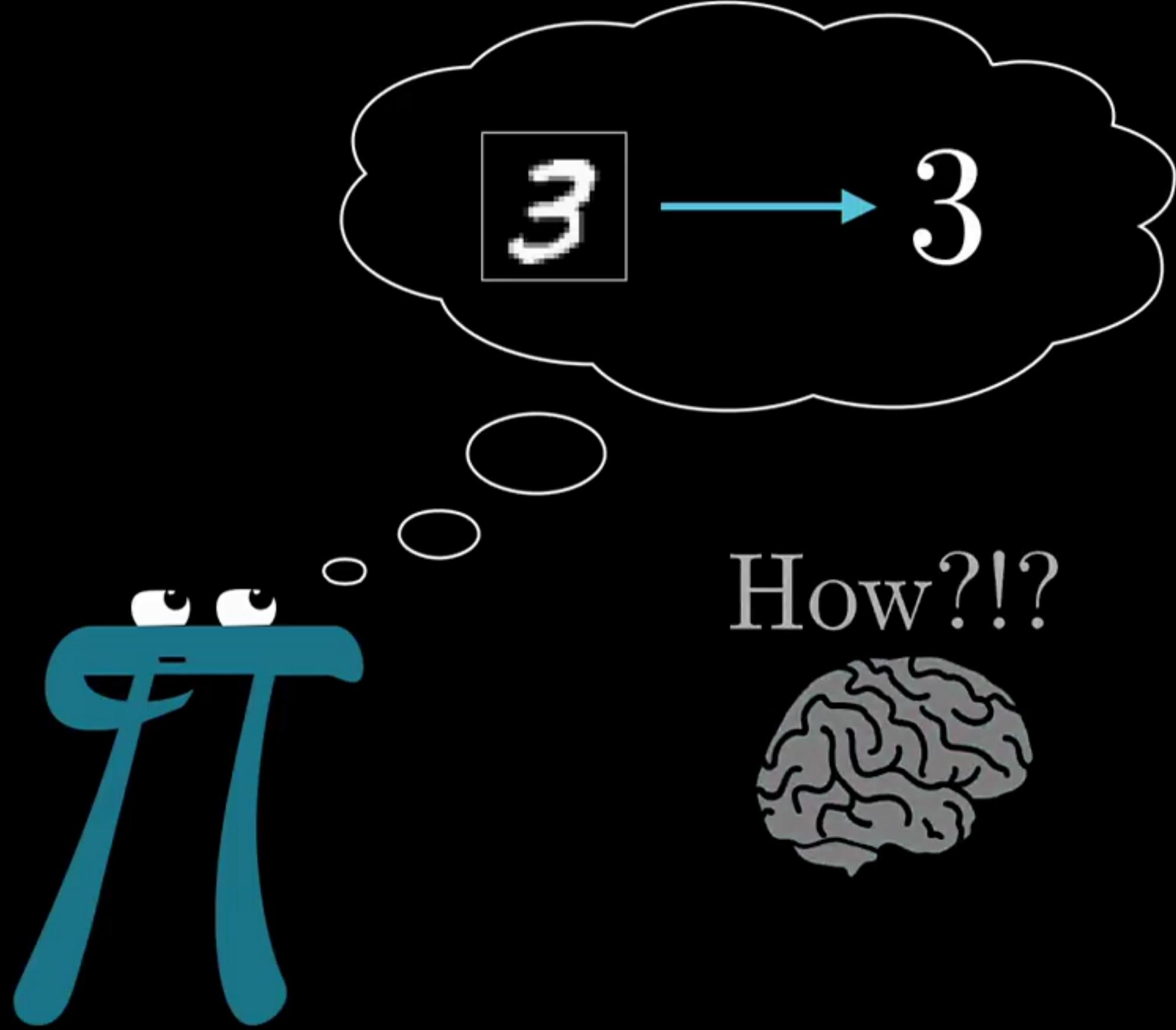


28px

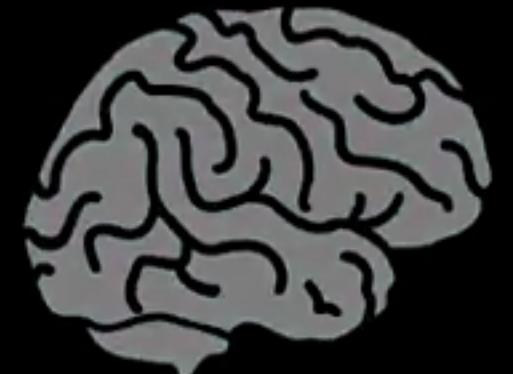


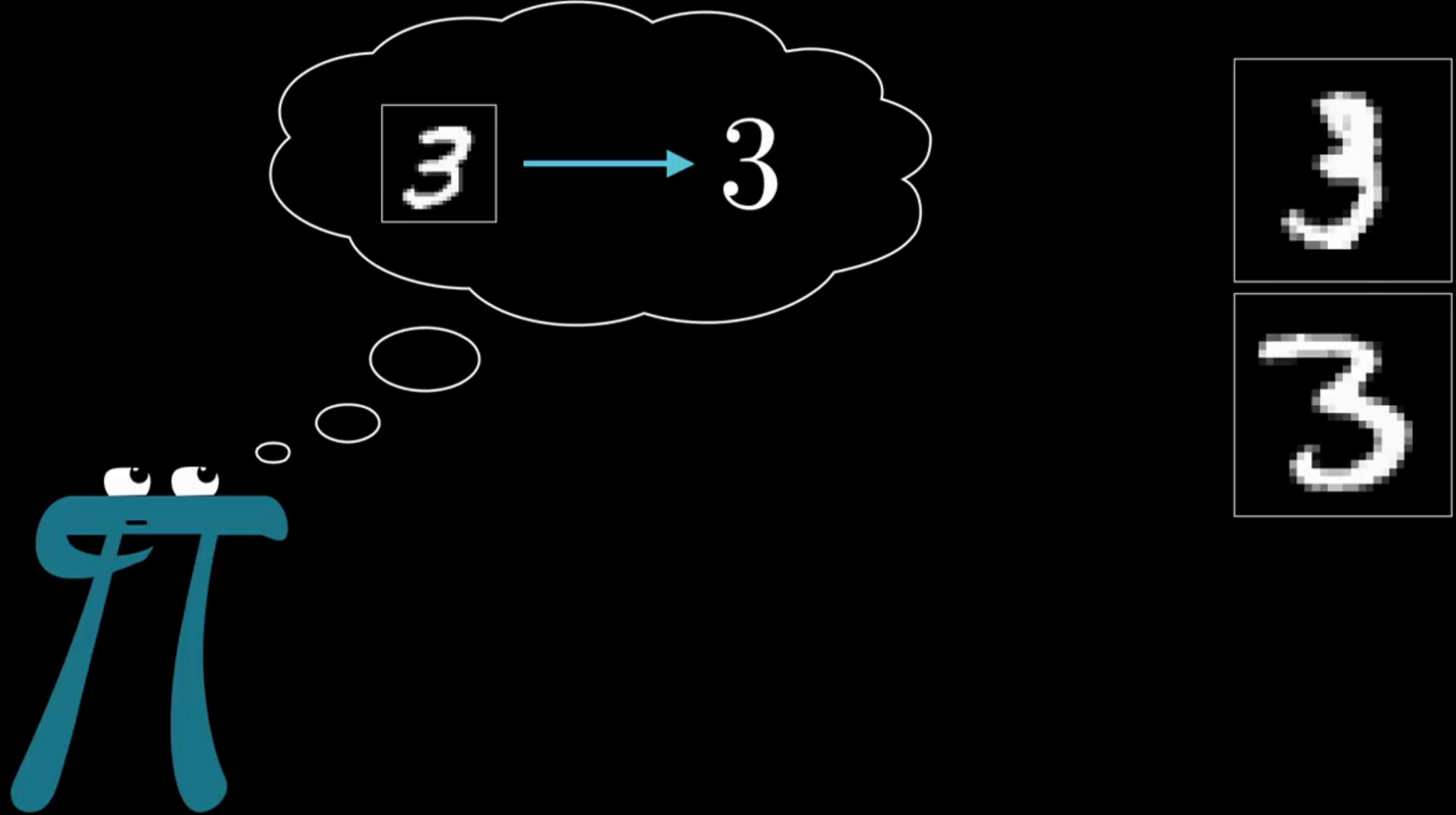
28px



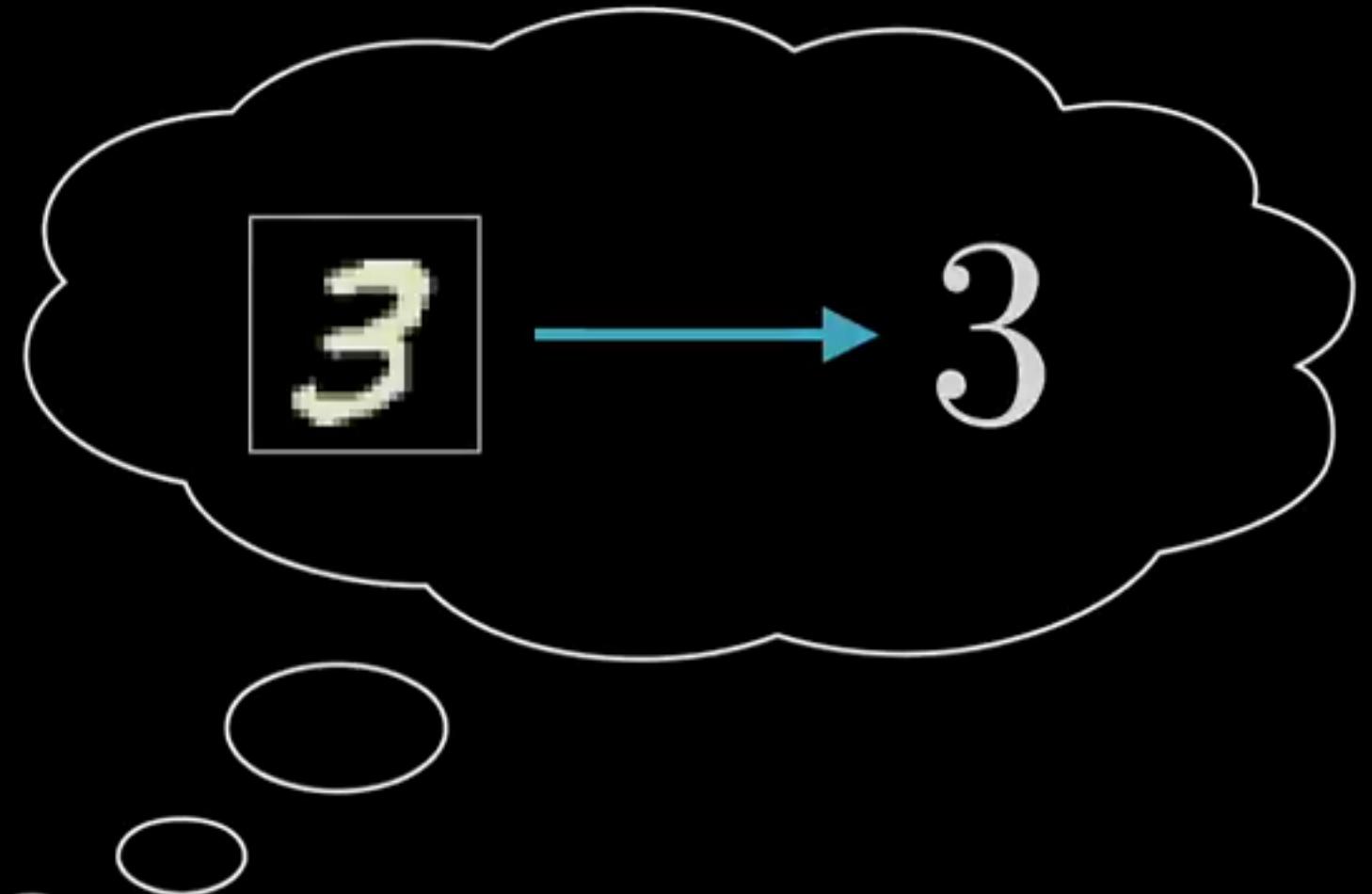


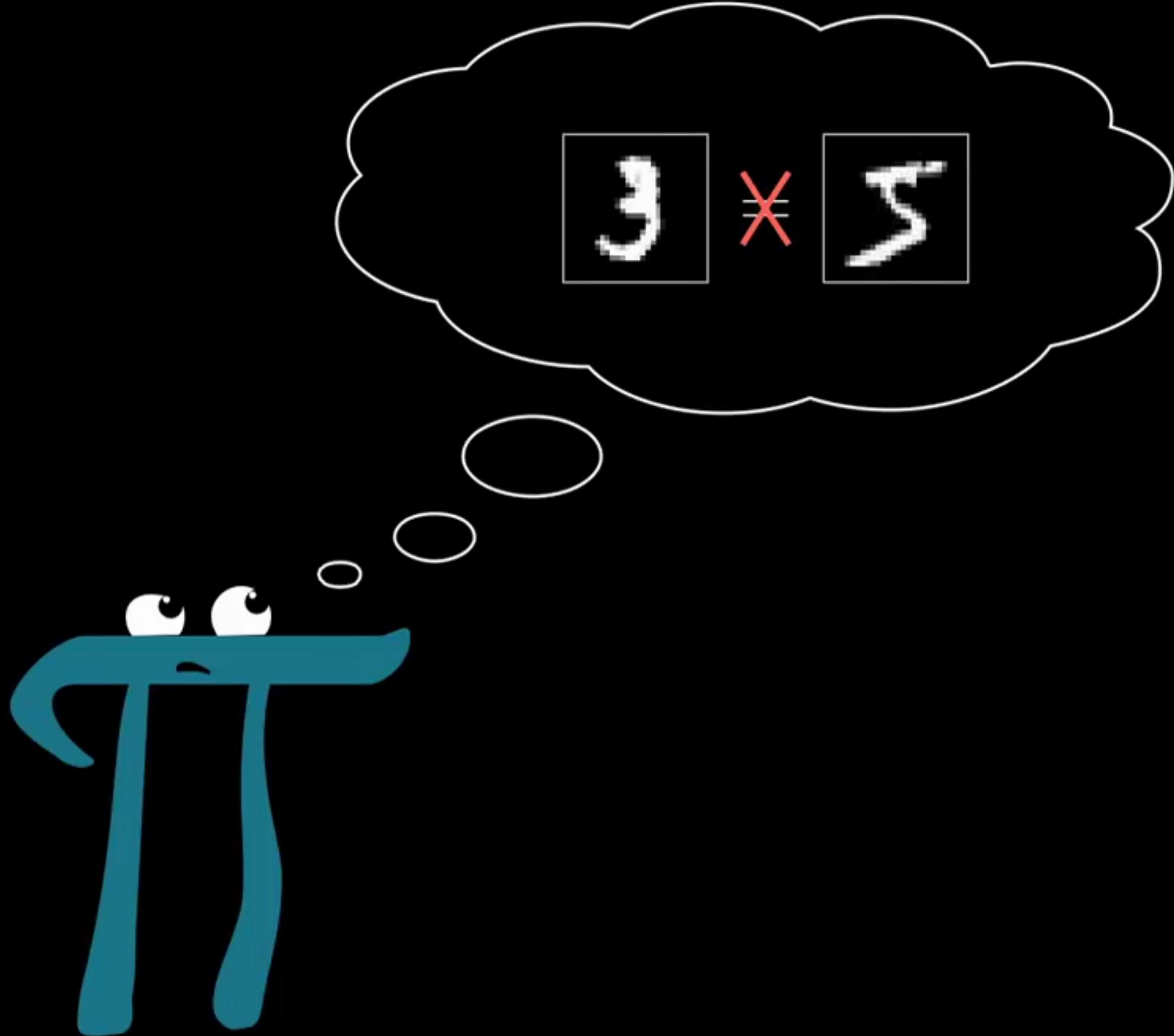
How?!?

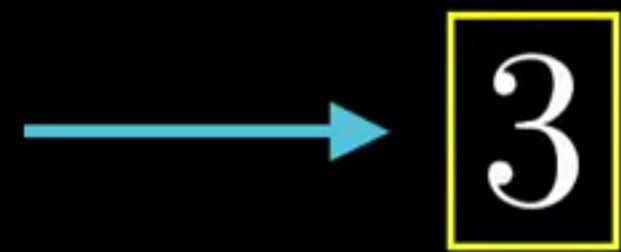




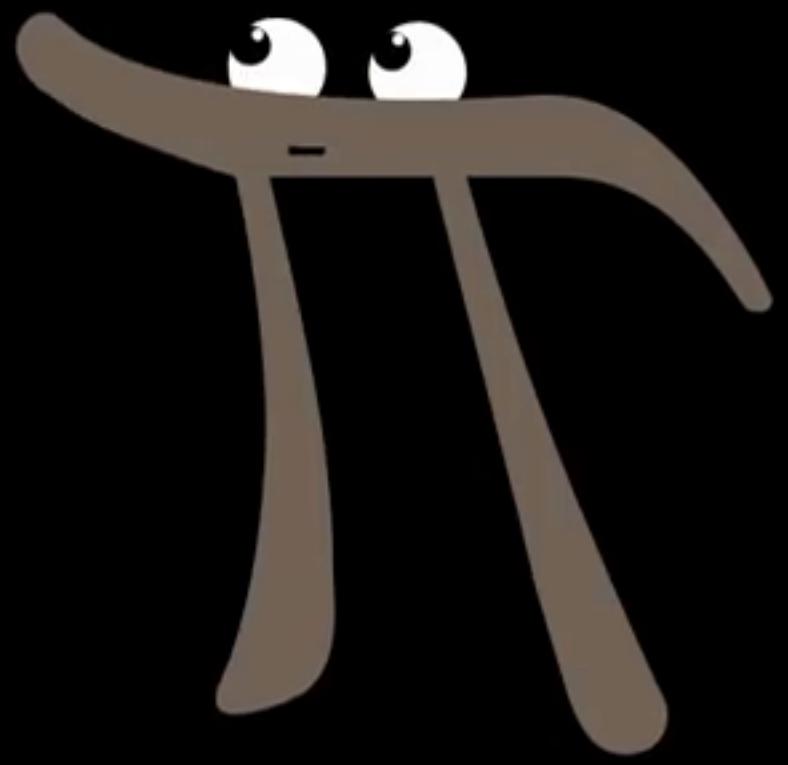
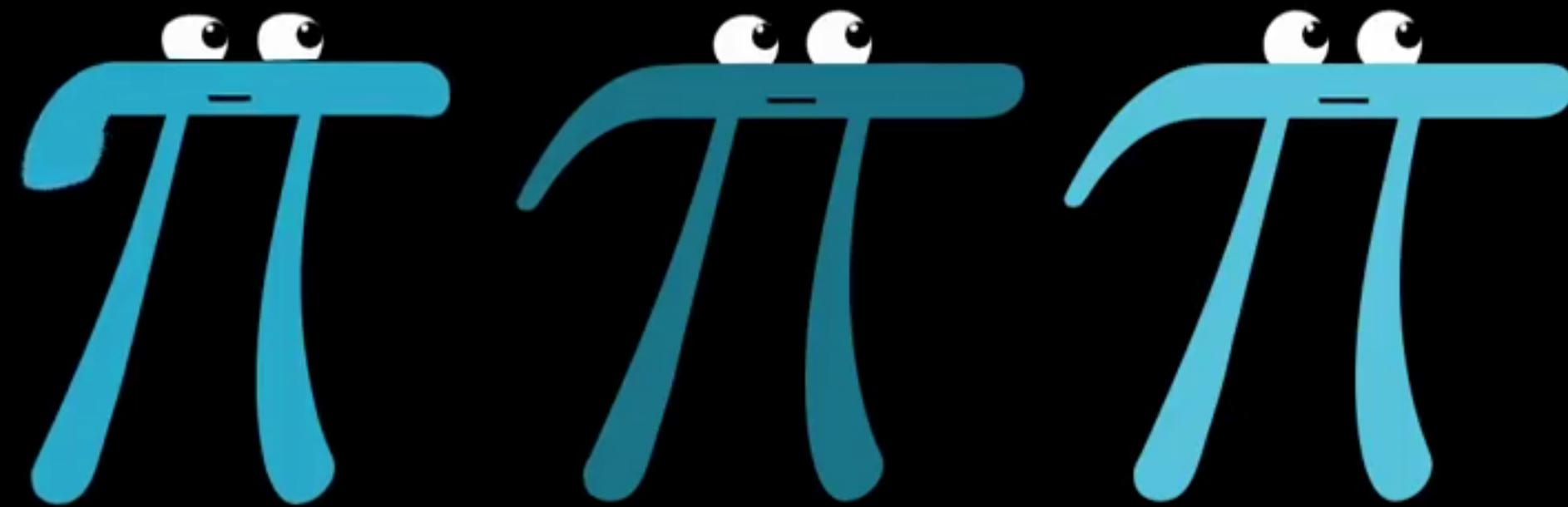
π



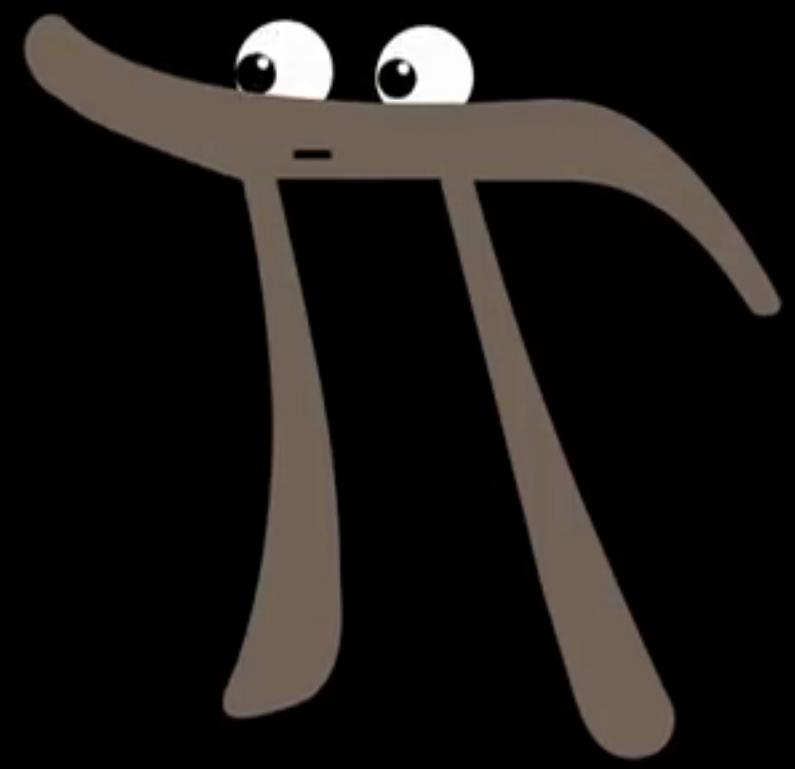
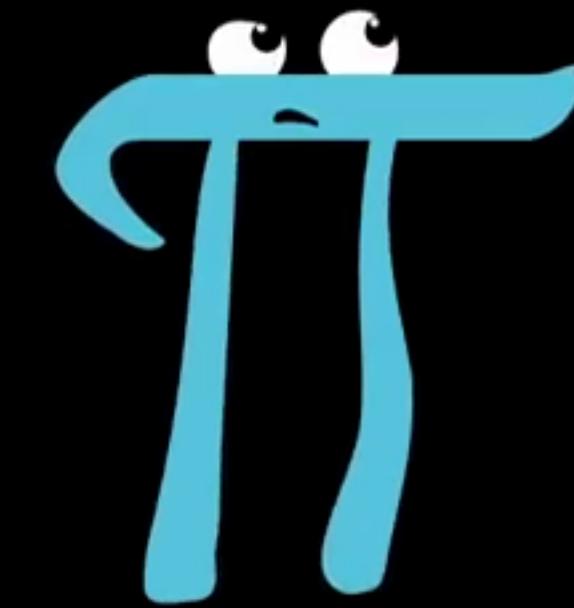
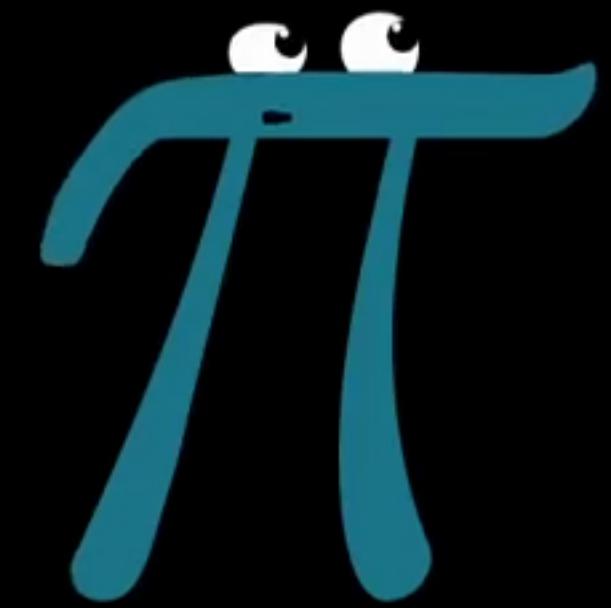
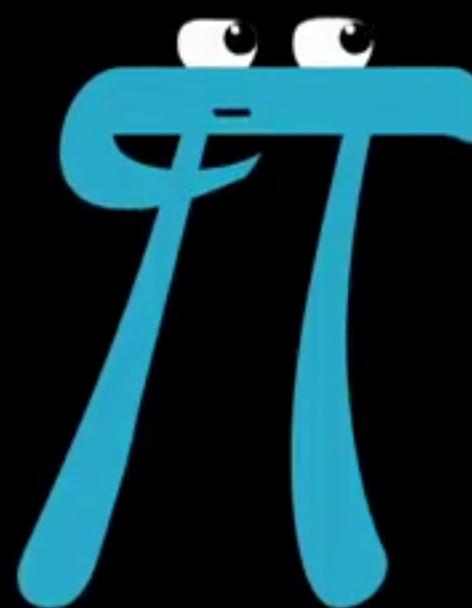




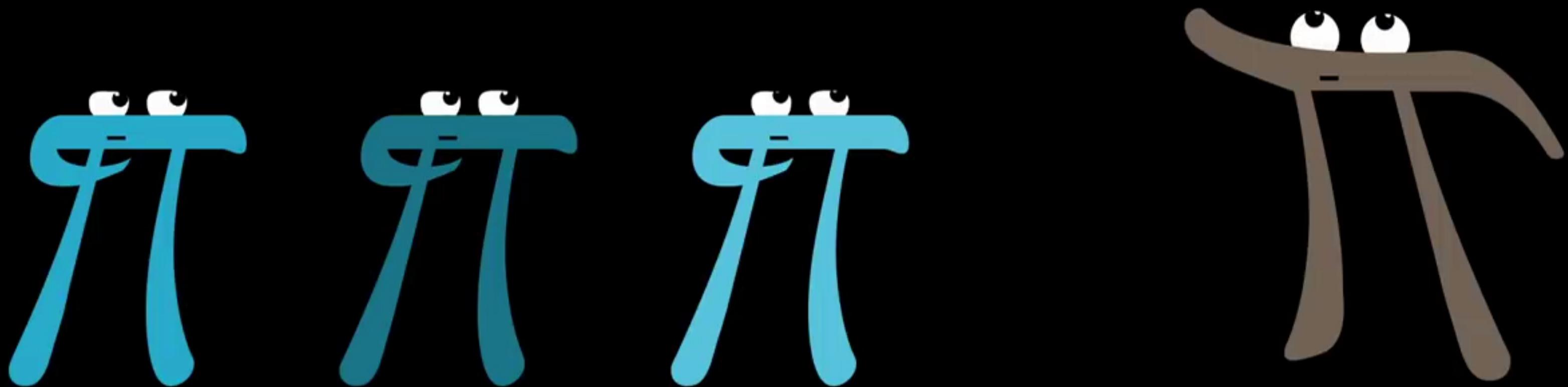
Machine learning



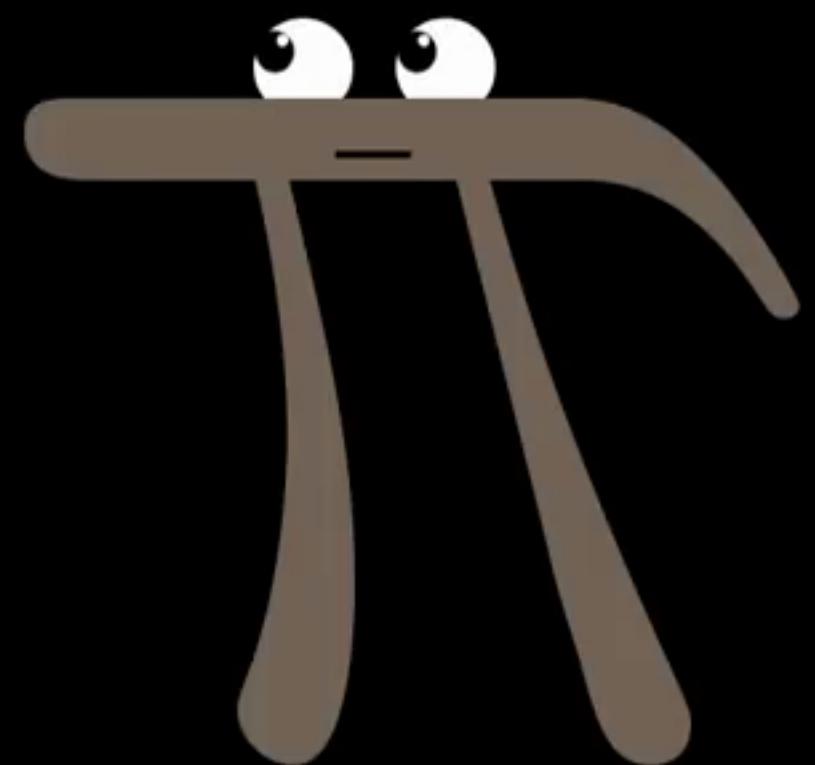
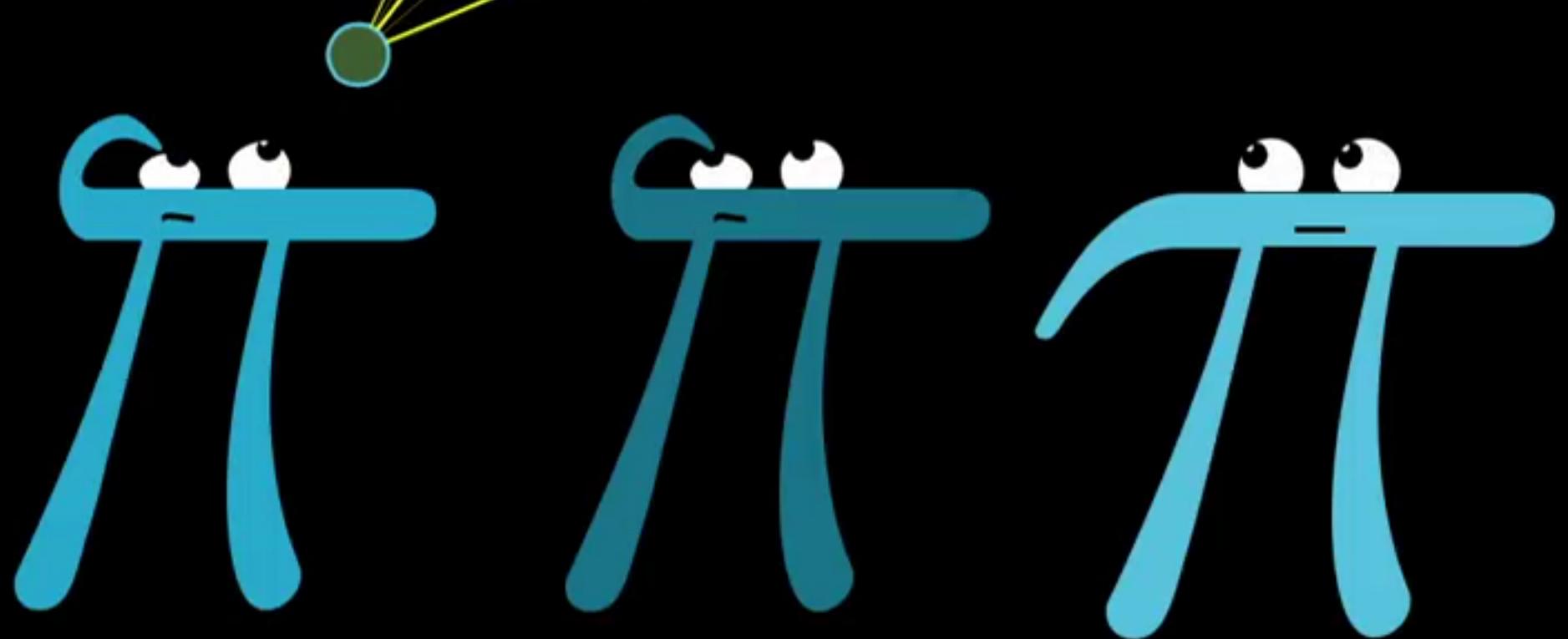
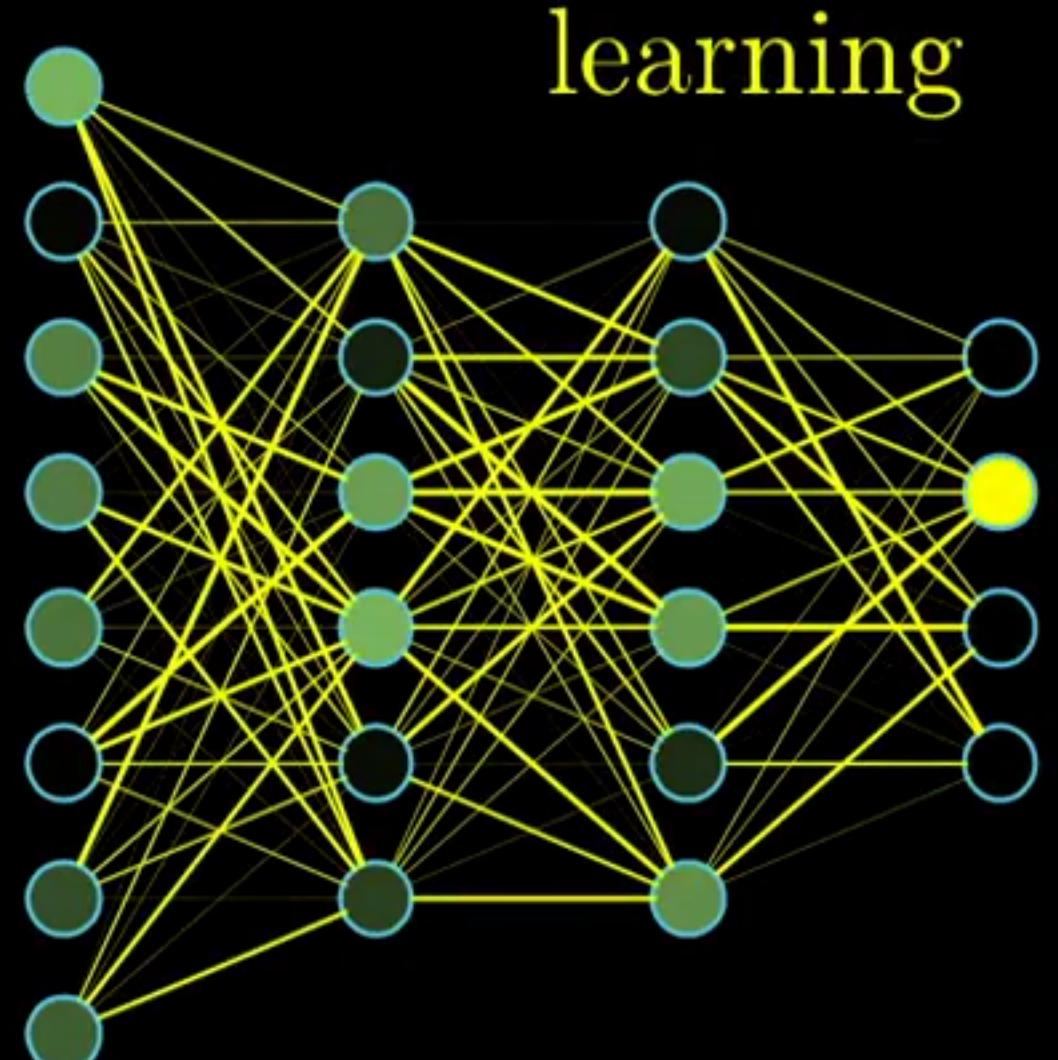
Machine learning
Neural network



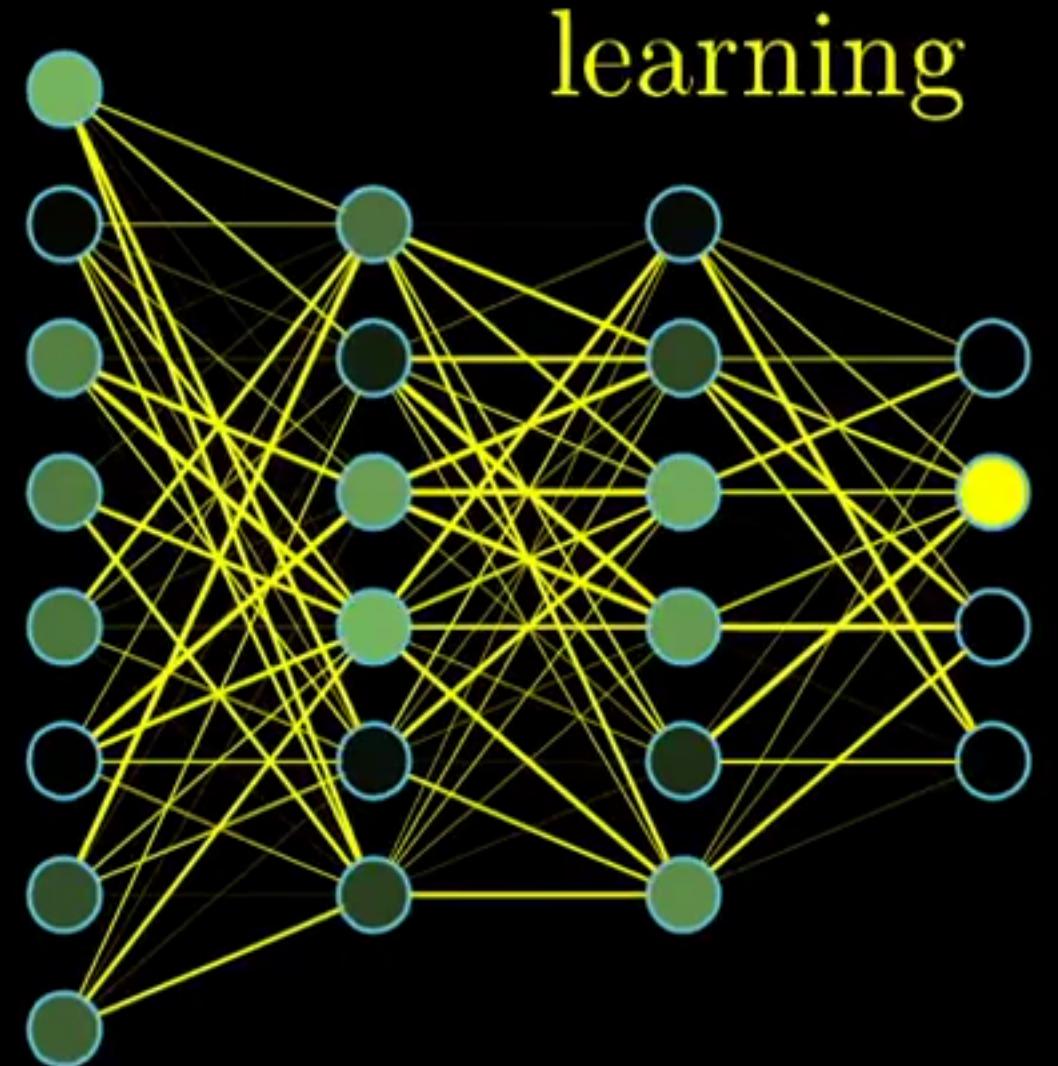
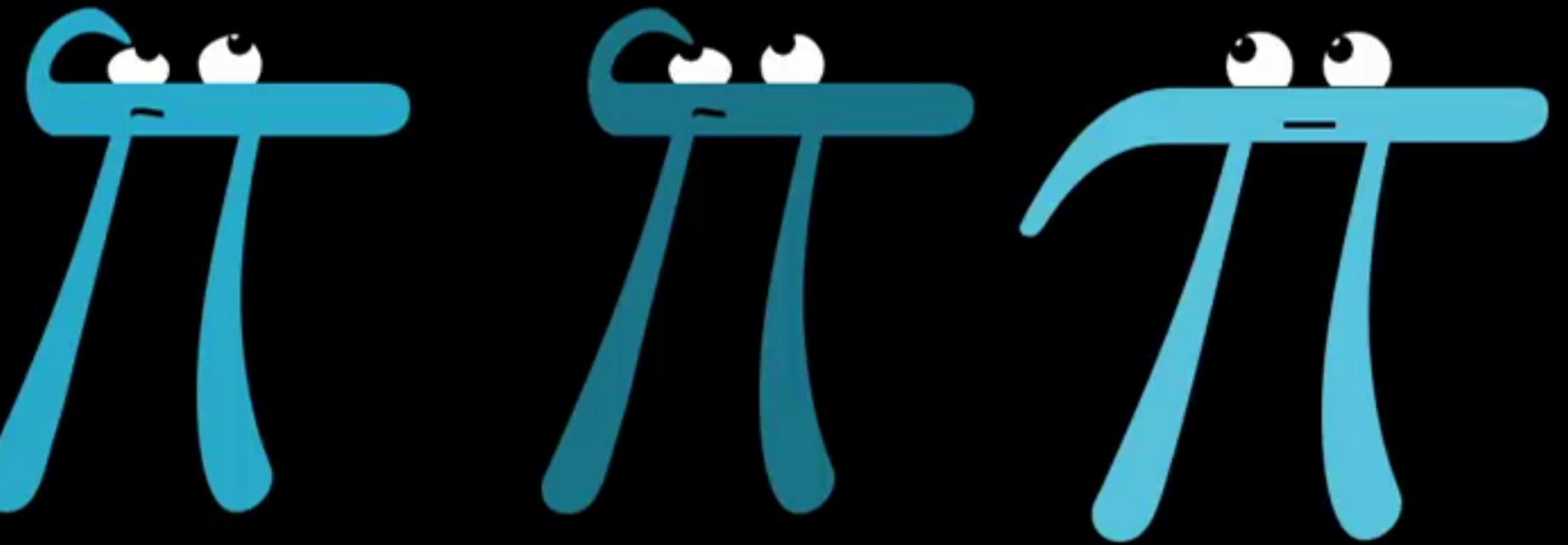
Machine learning
Neural network



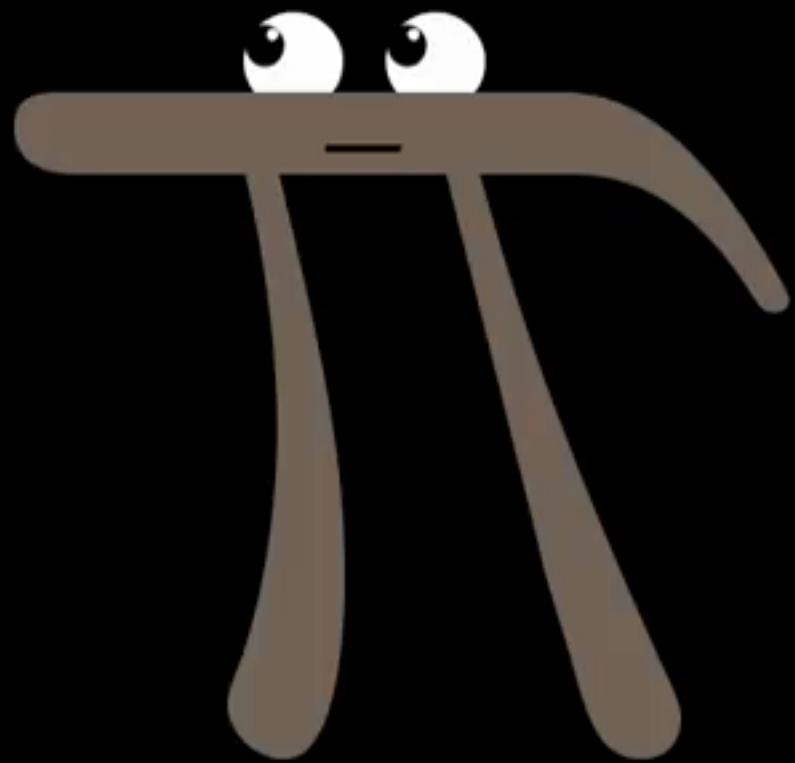
Machine learning
Neural network



learning

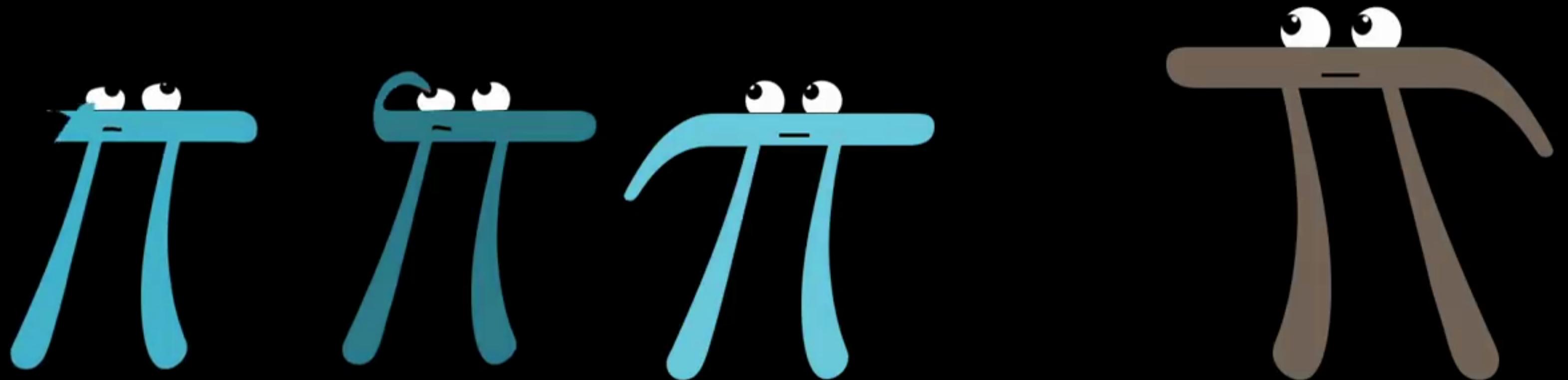
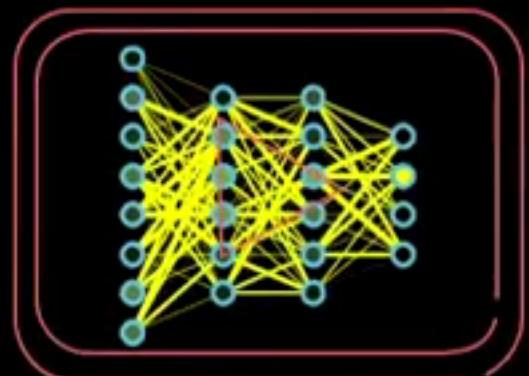


Machine learning
Neural network



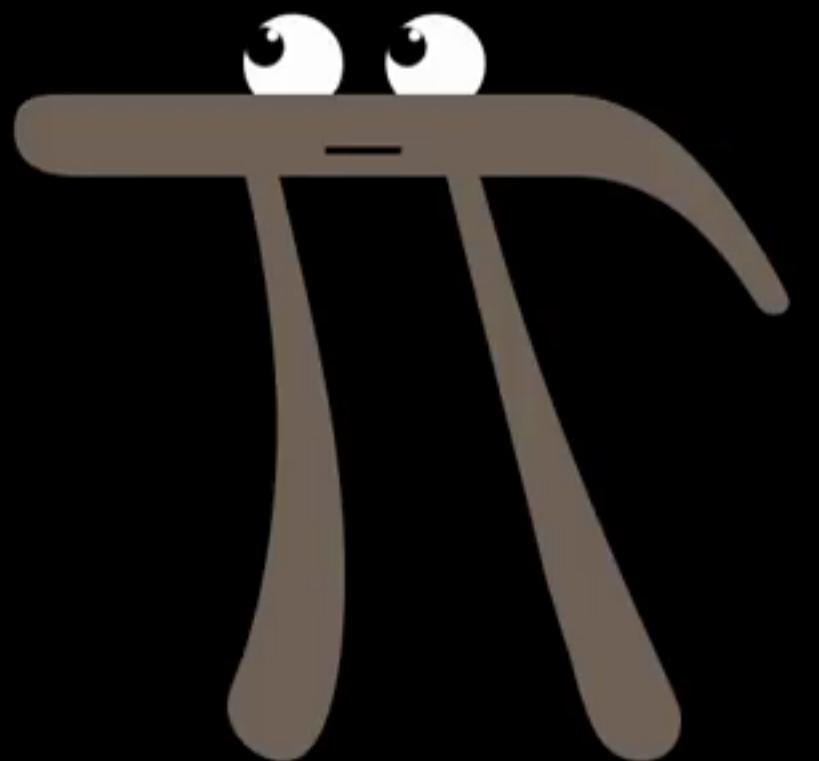
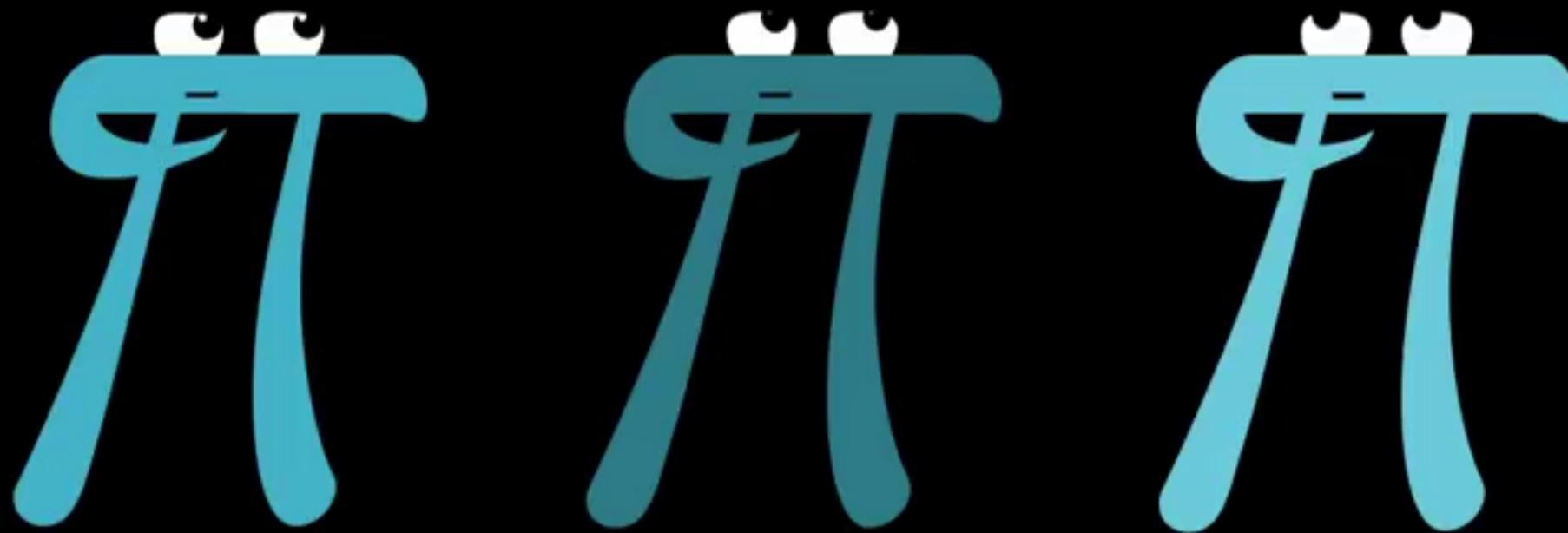
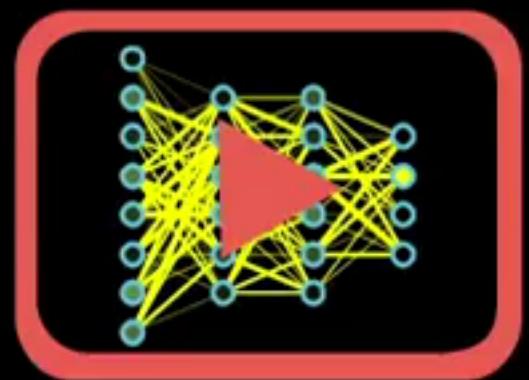
Machine learning
Neural network

Structure
learning



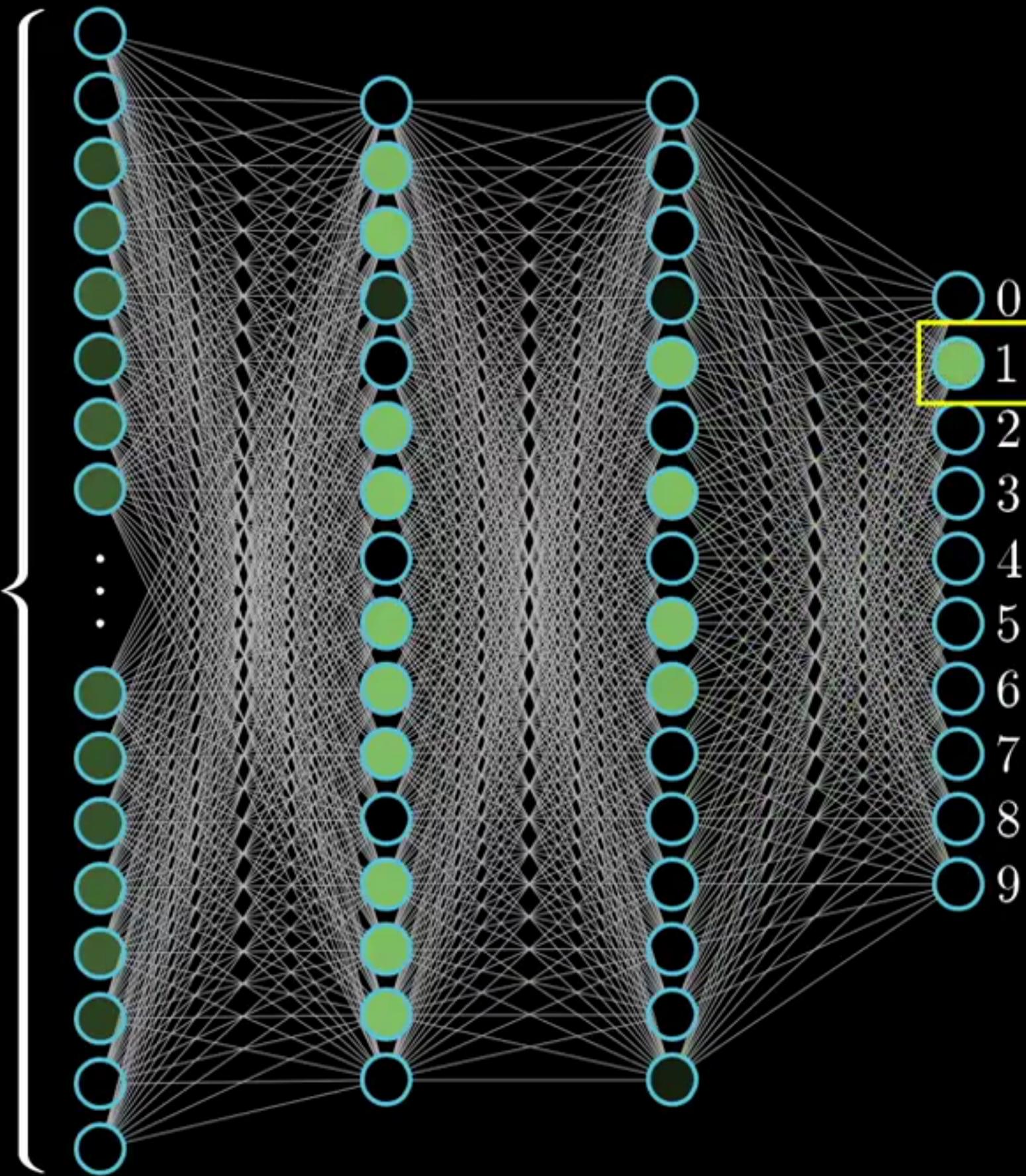
Machine learning
Neural network

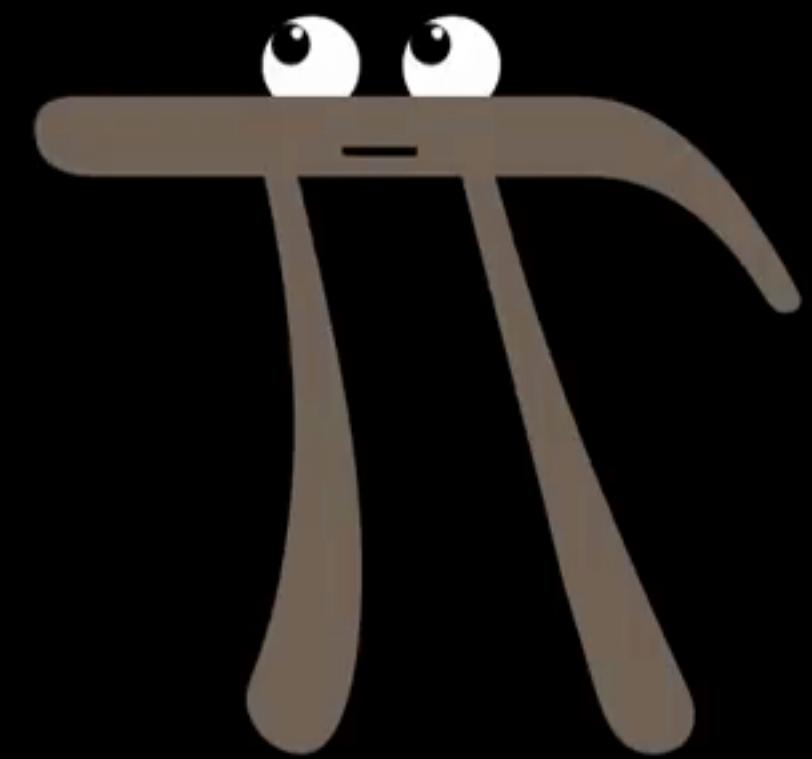
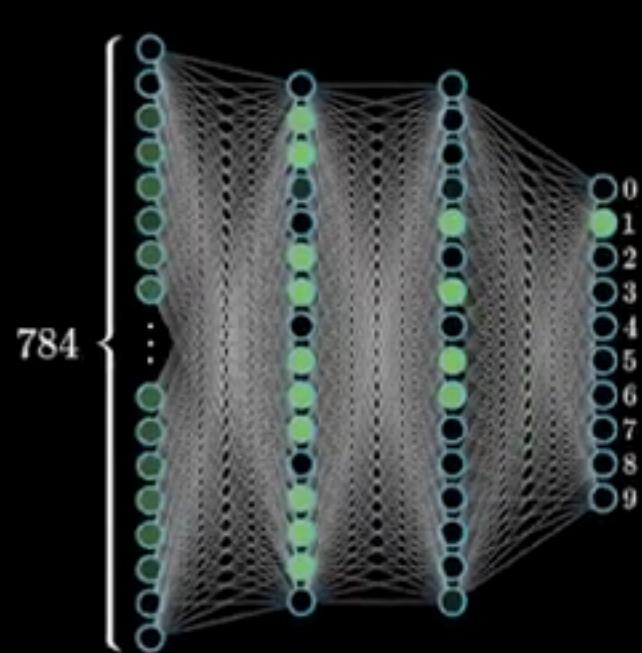
Structure learning

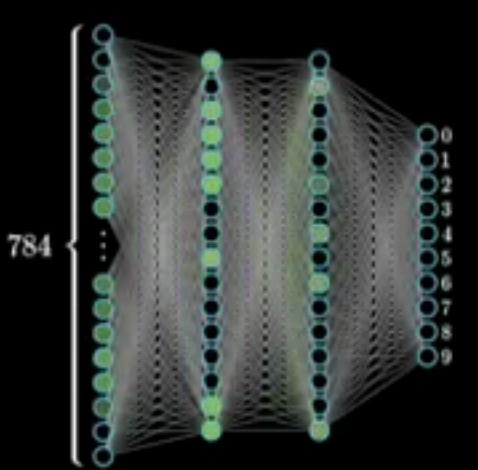
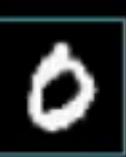




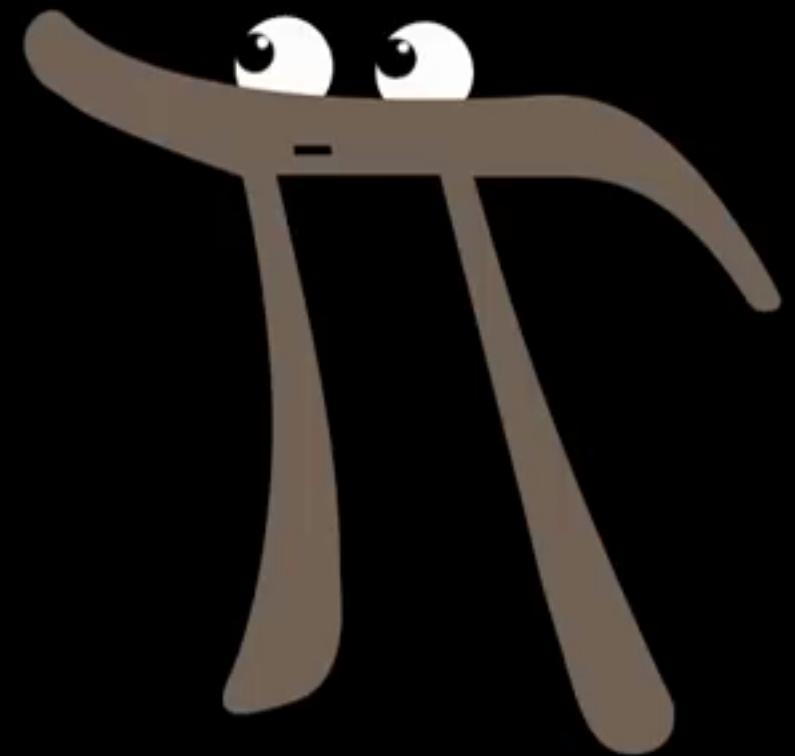
784

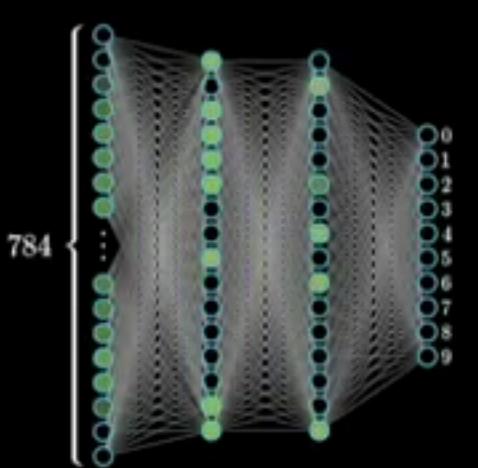






Long Convolutional neural network → Good for image recognition

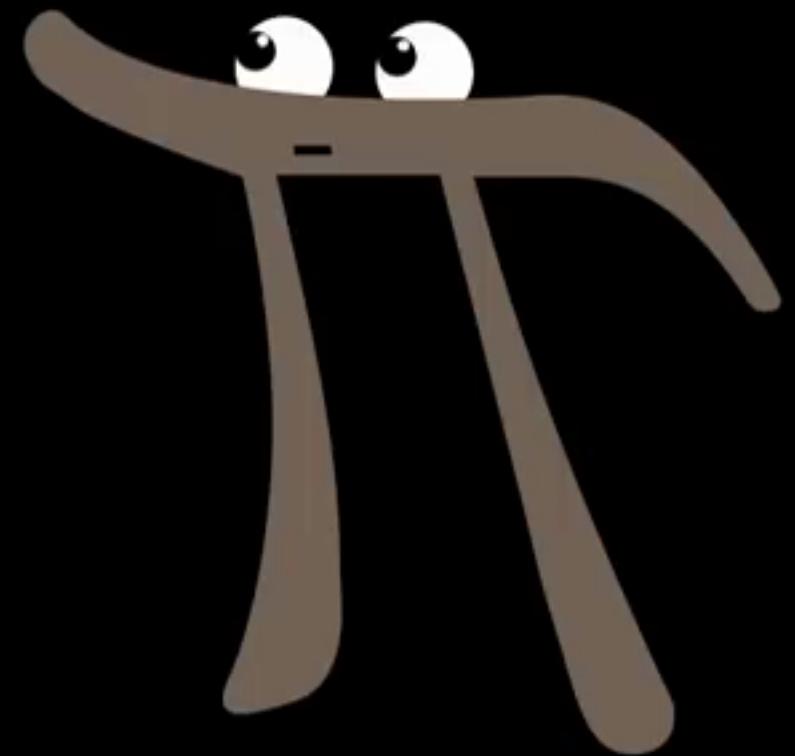


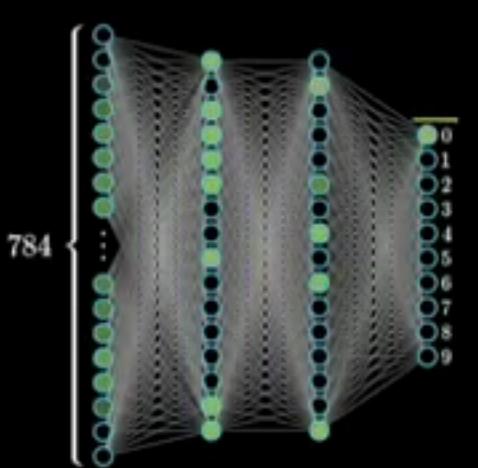


Convolutional neural network
Long short-term memory network

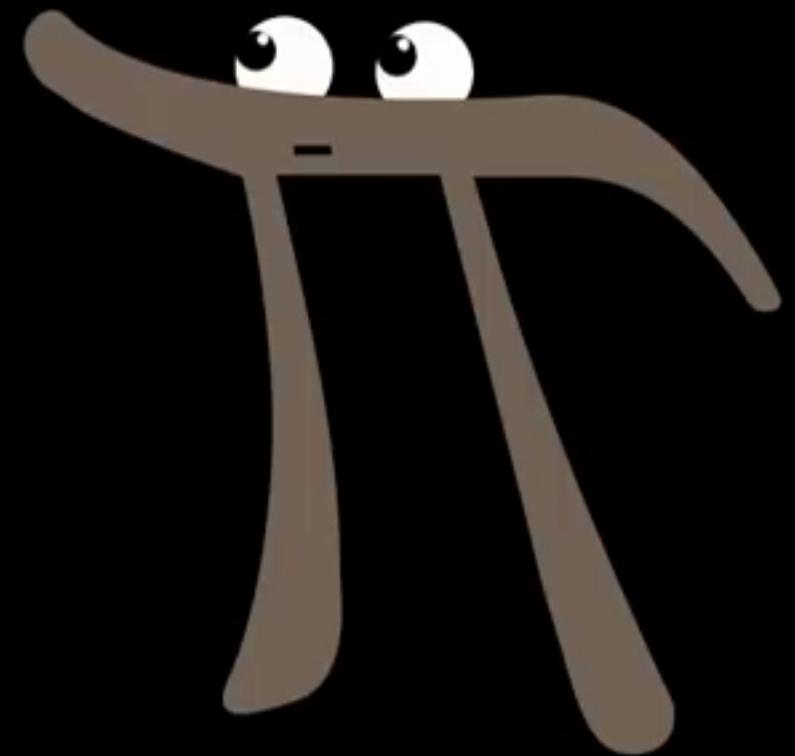


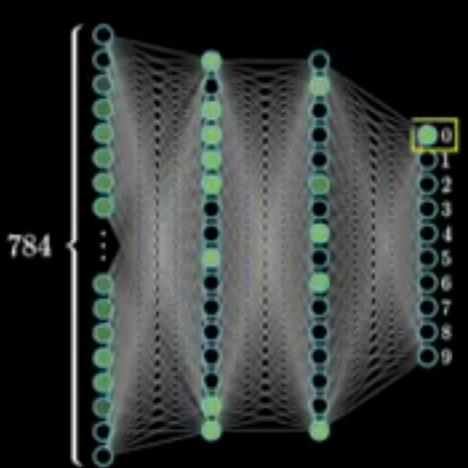
Good for image recognition
Good for speech recognition



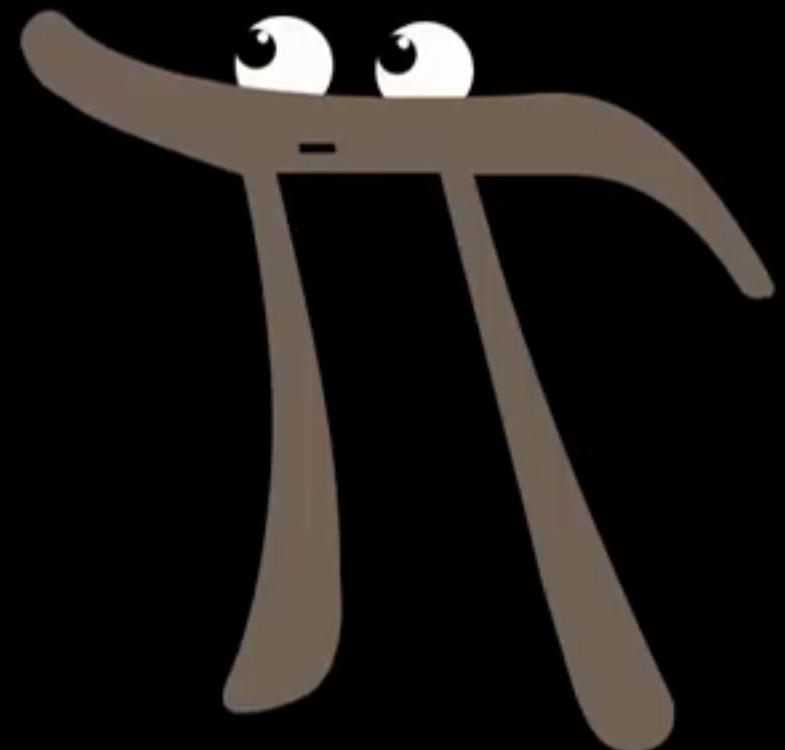


Convolutional neural network → Good for image recognition
Long short-term memory network → Good for speech recognition



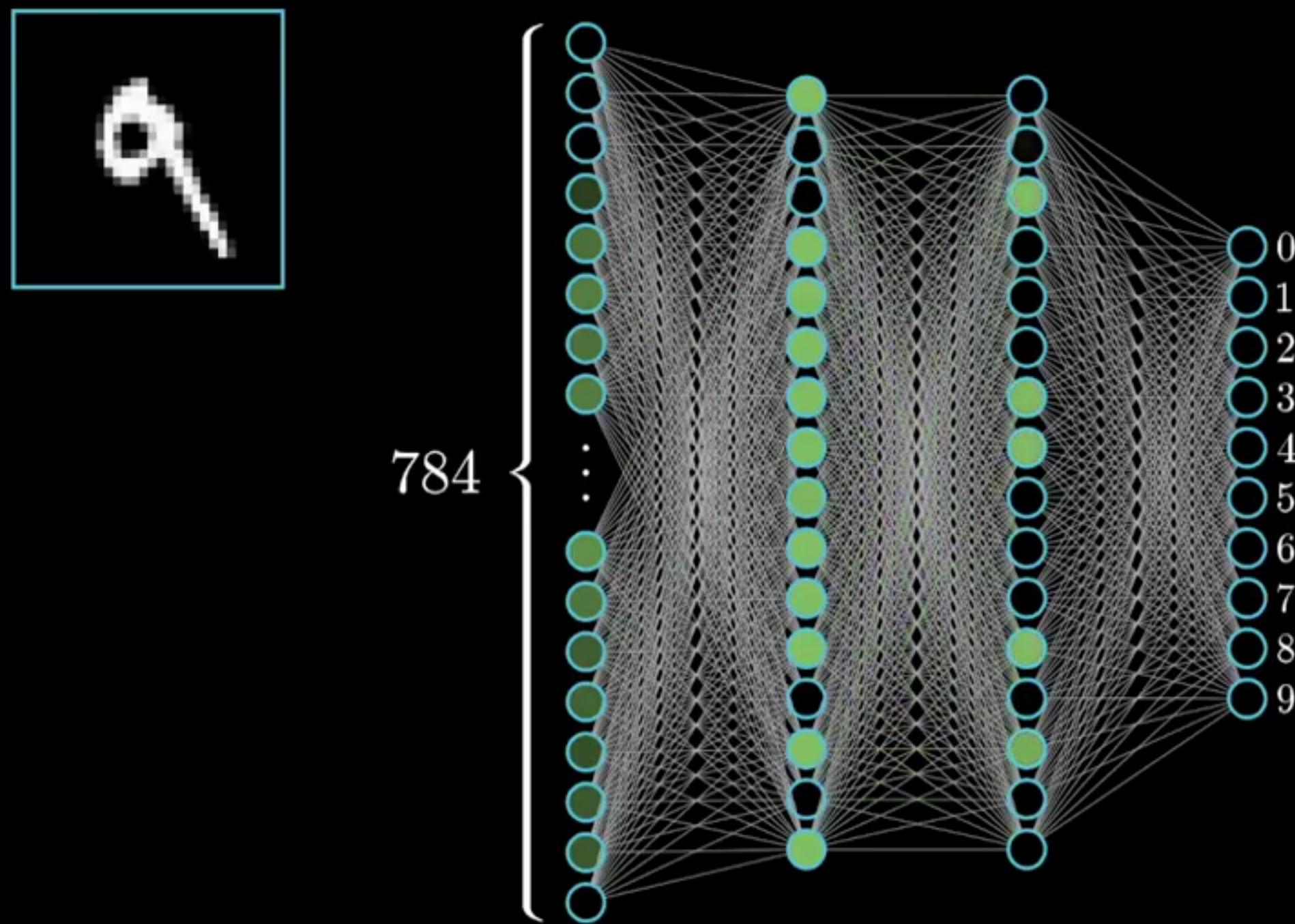


Convolutional neural network → Good for image recognition
Long short-term memory network → Good for speech recognition



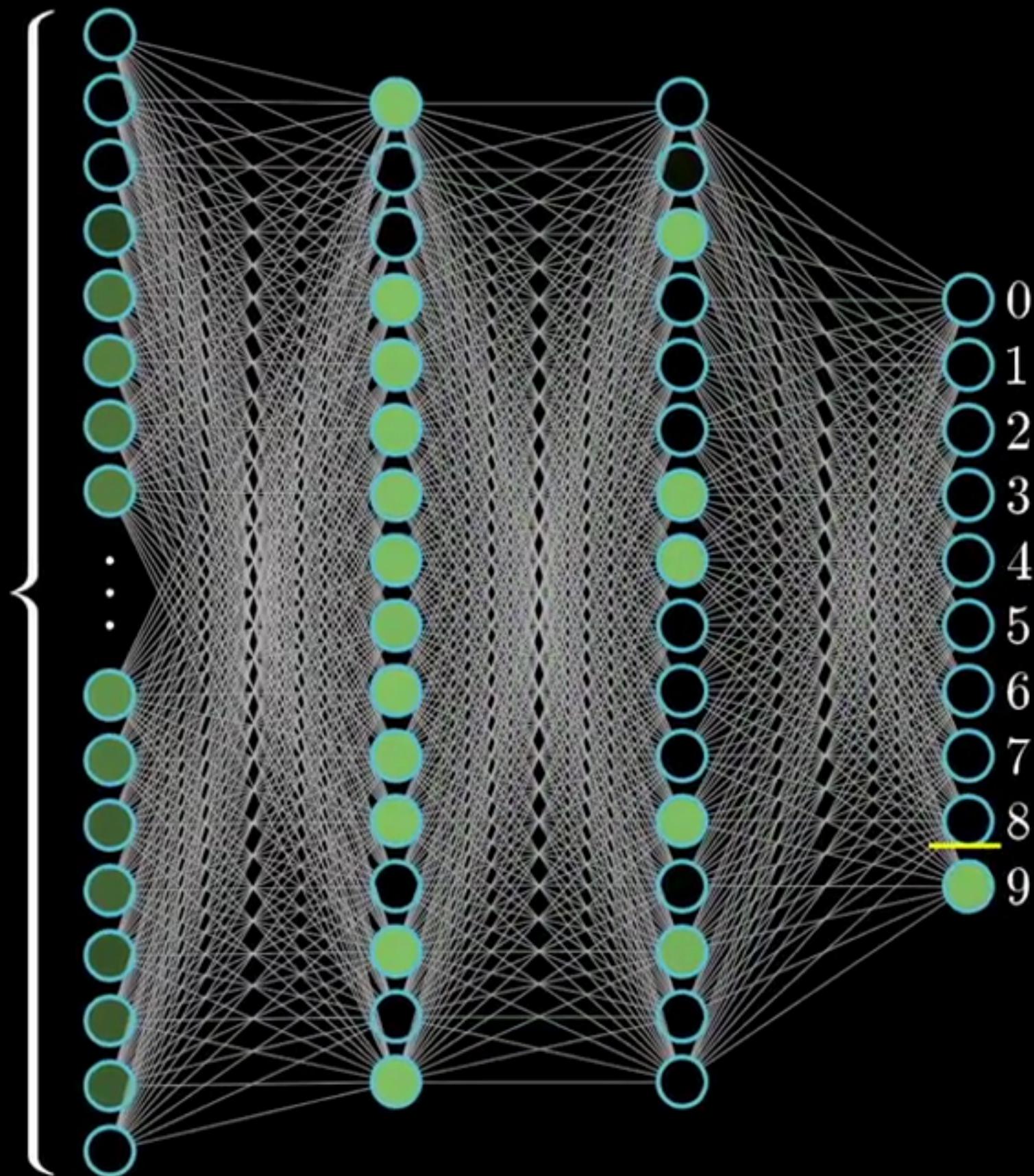
Plain vanilla

(aka “multilayer perceptron”)



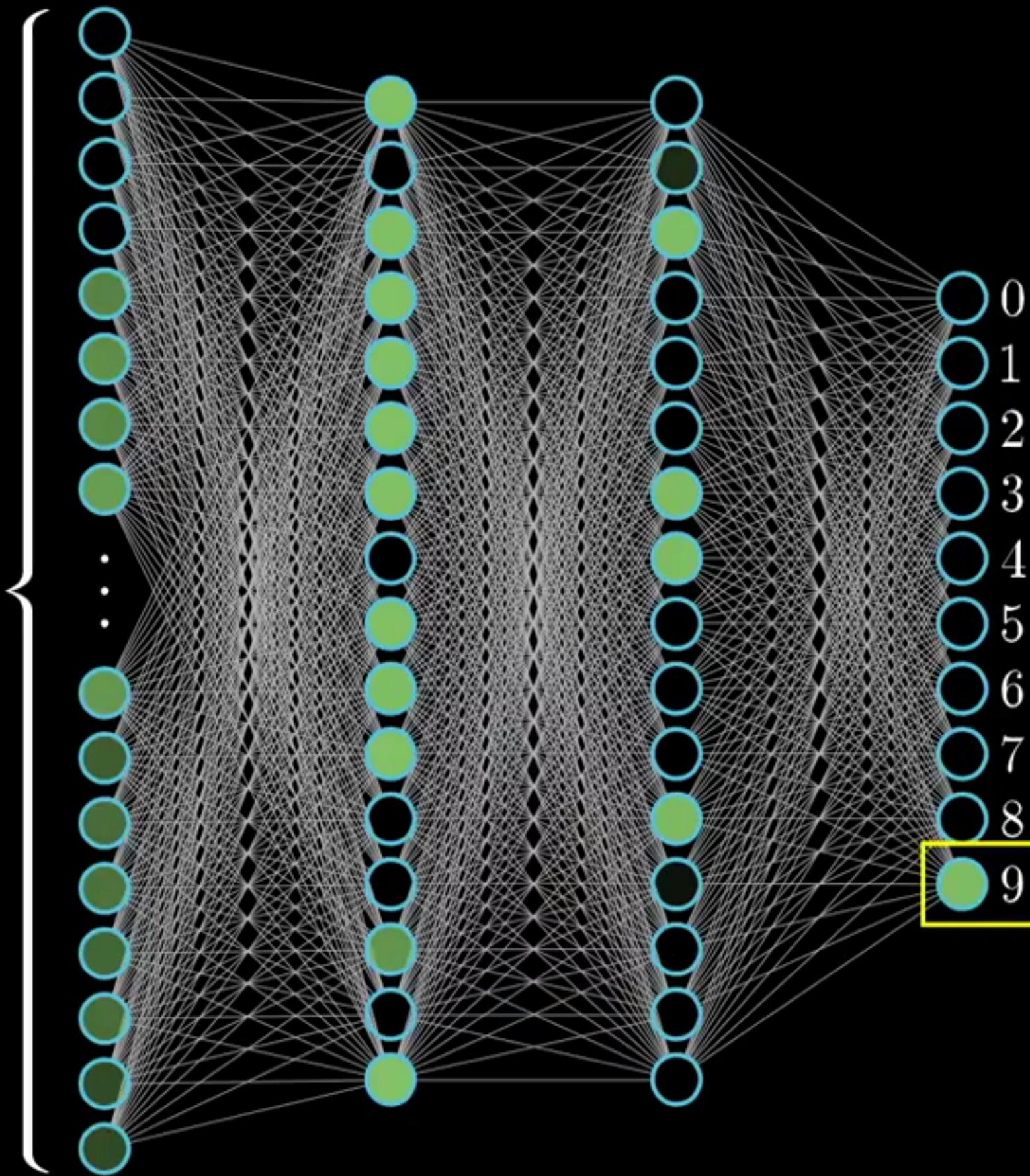


784

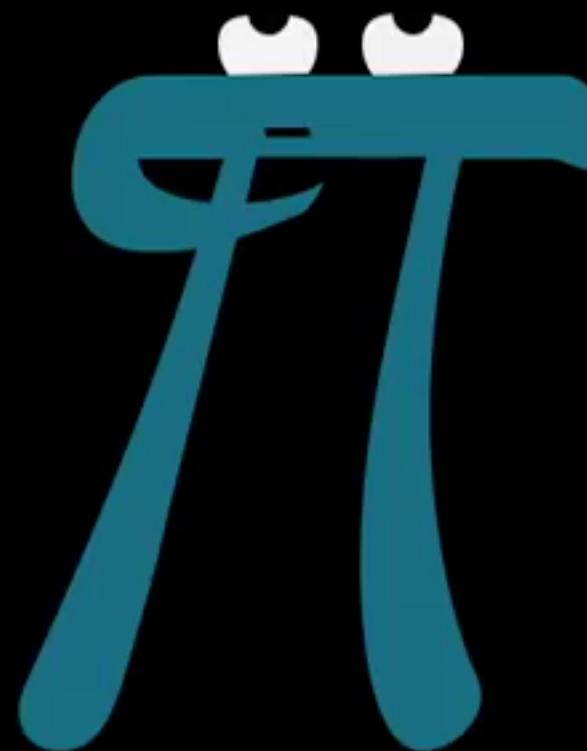




784



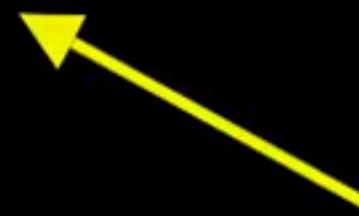
- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



Neural network



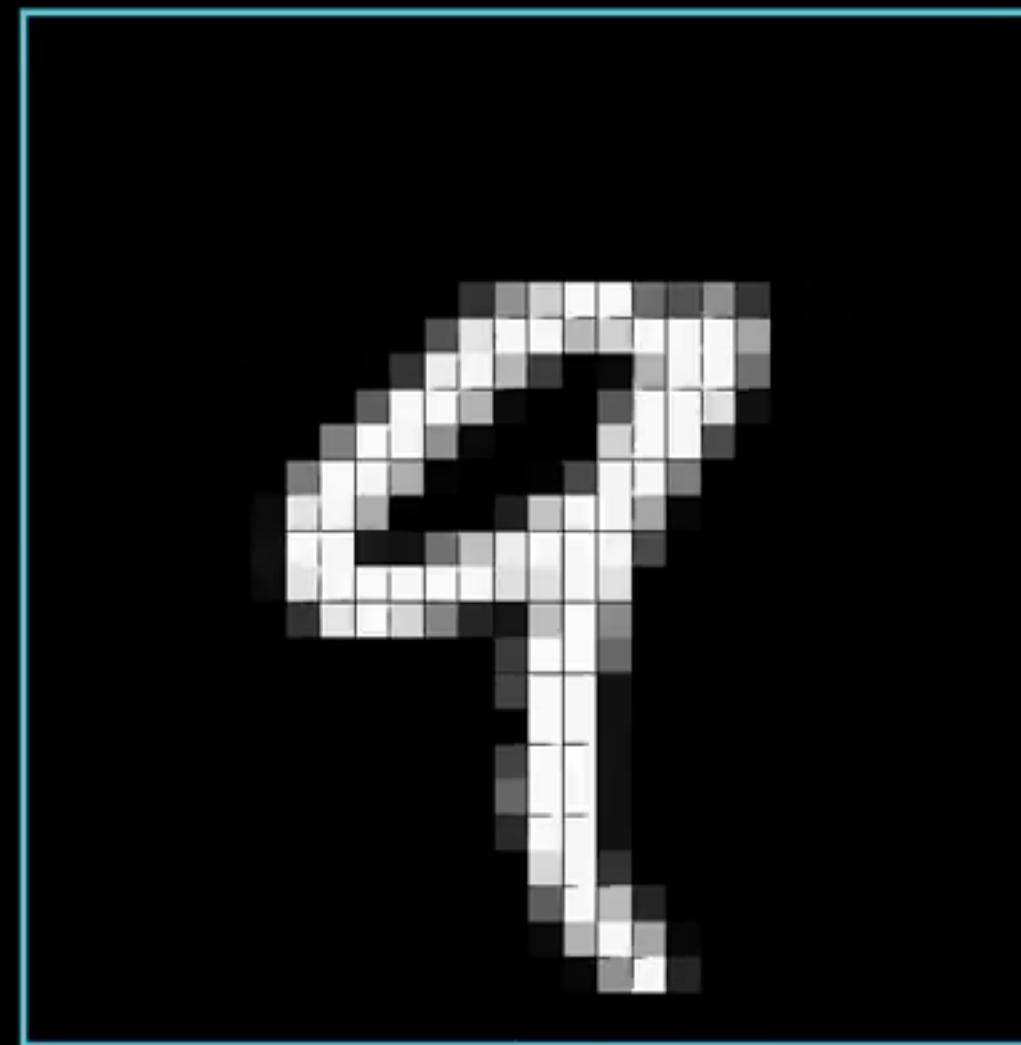
What are
the neurons?



How are
they connected?

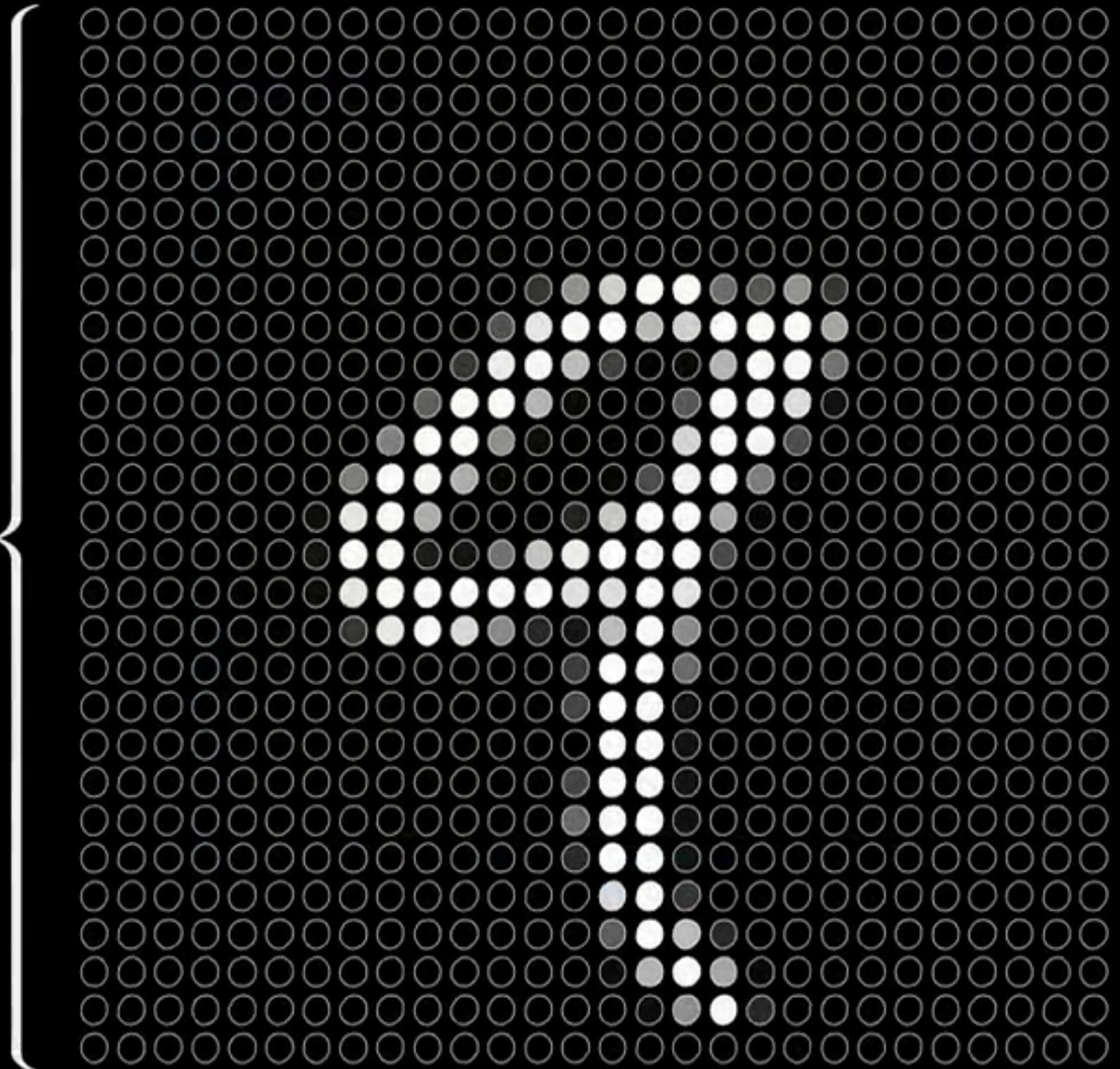
0.1

Neuron → Thing that holds a number



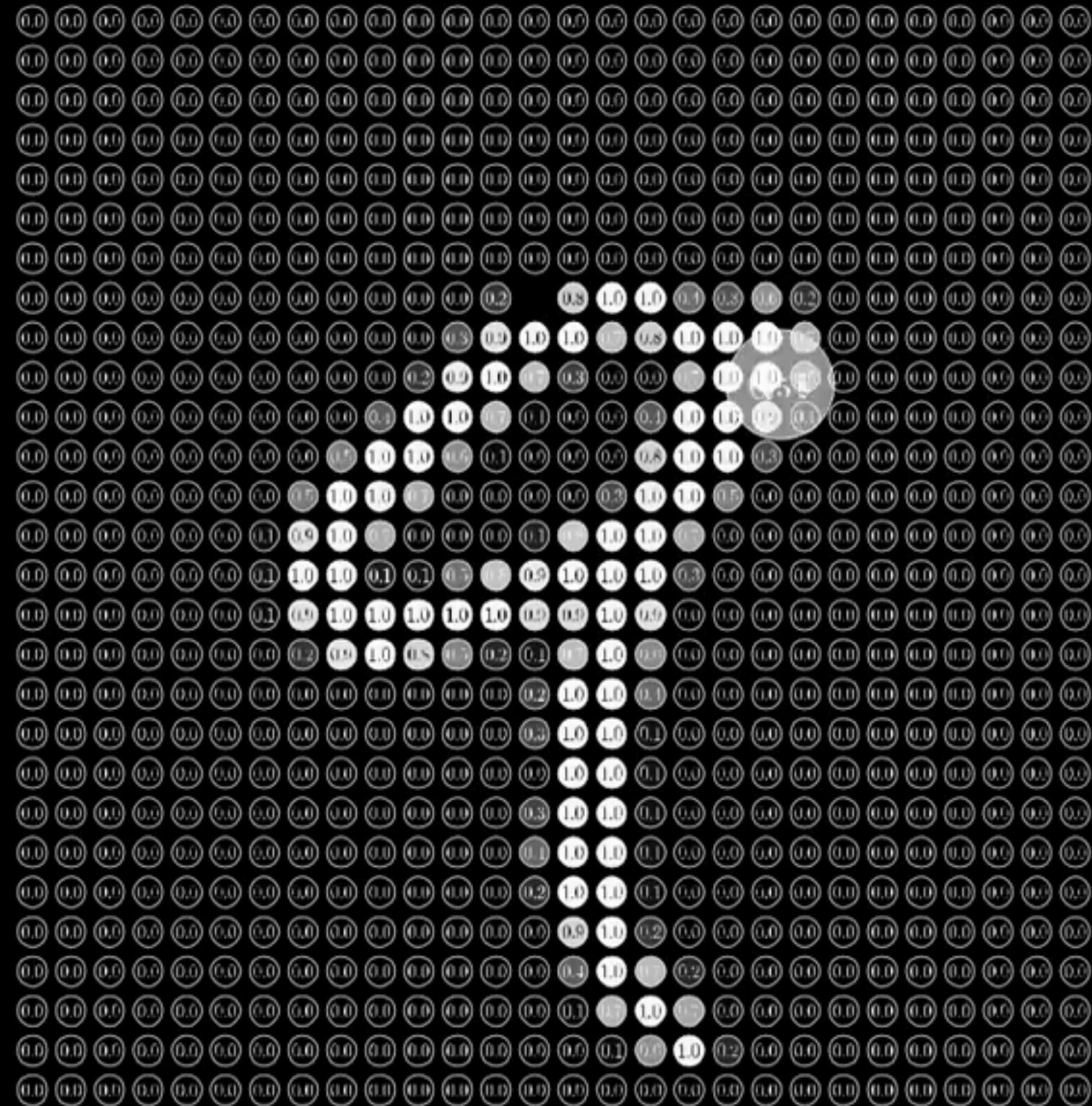
28

28



28

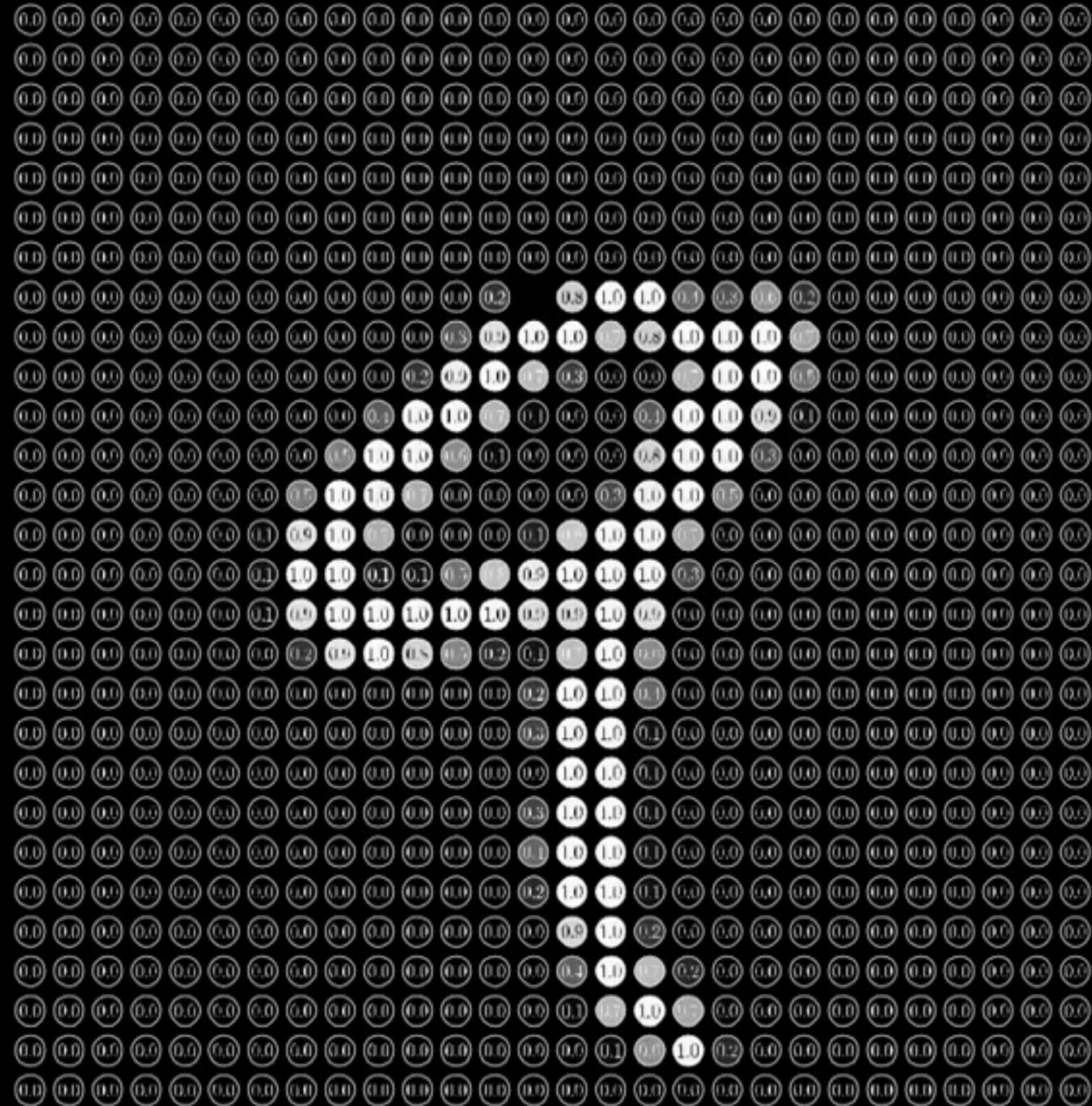
28 {



$$28 \times 28 = 784$$

28

28

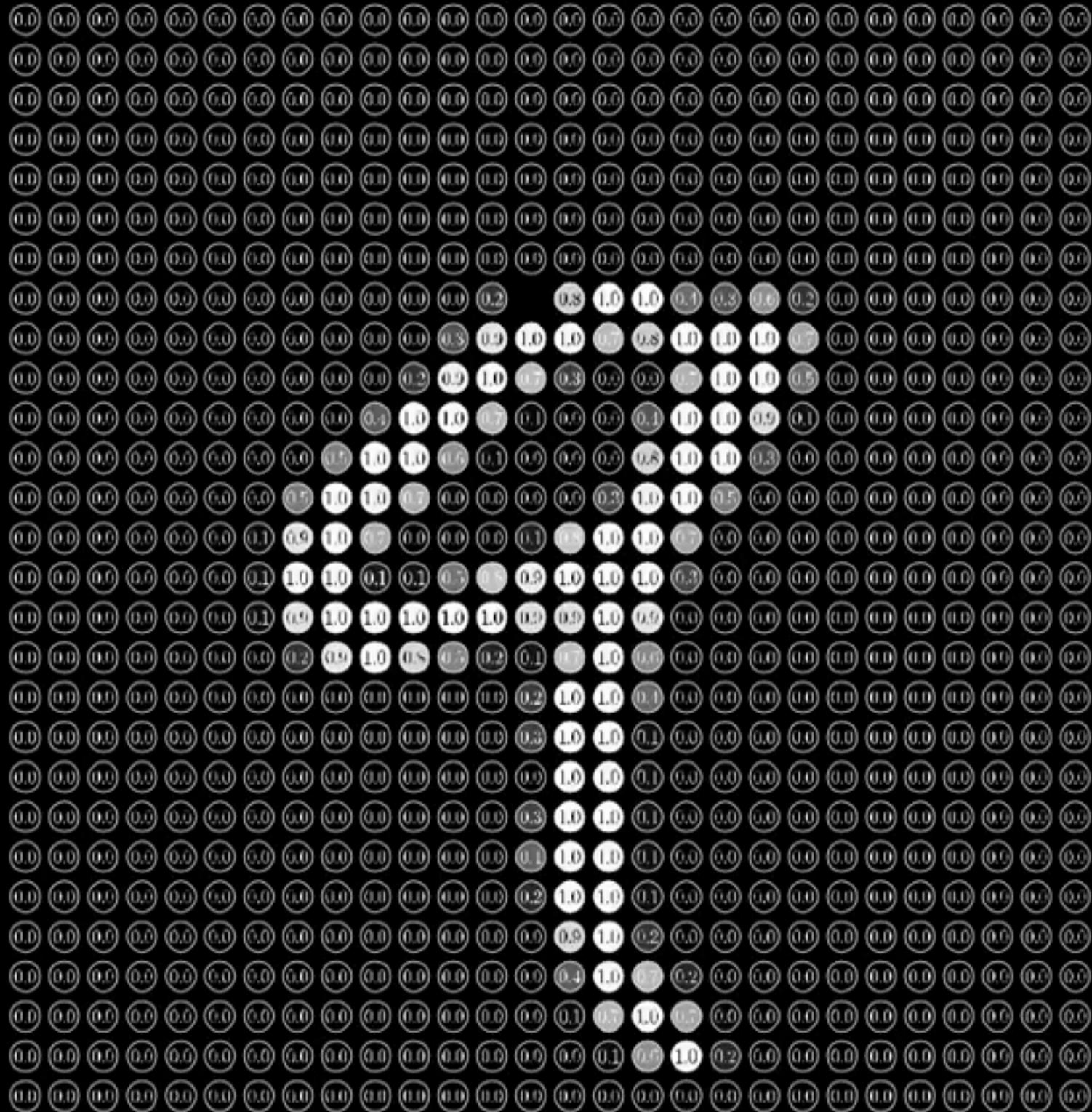


$$28 \times 28 = 784$$

0.58

28

28

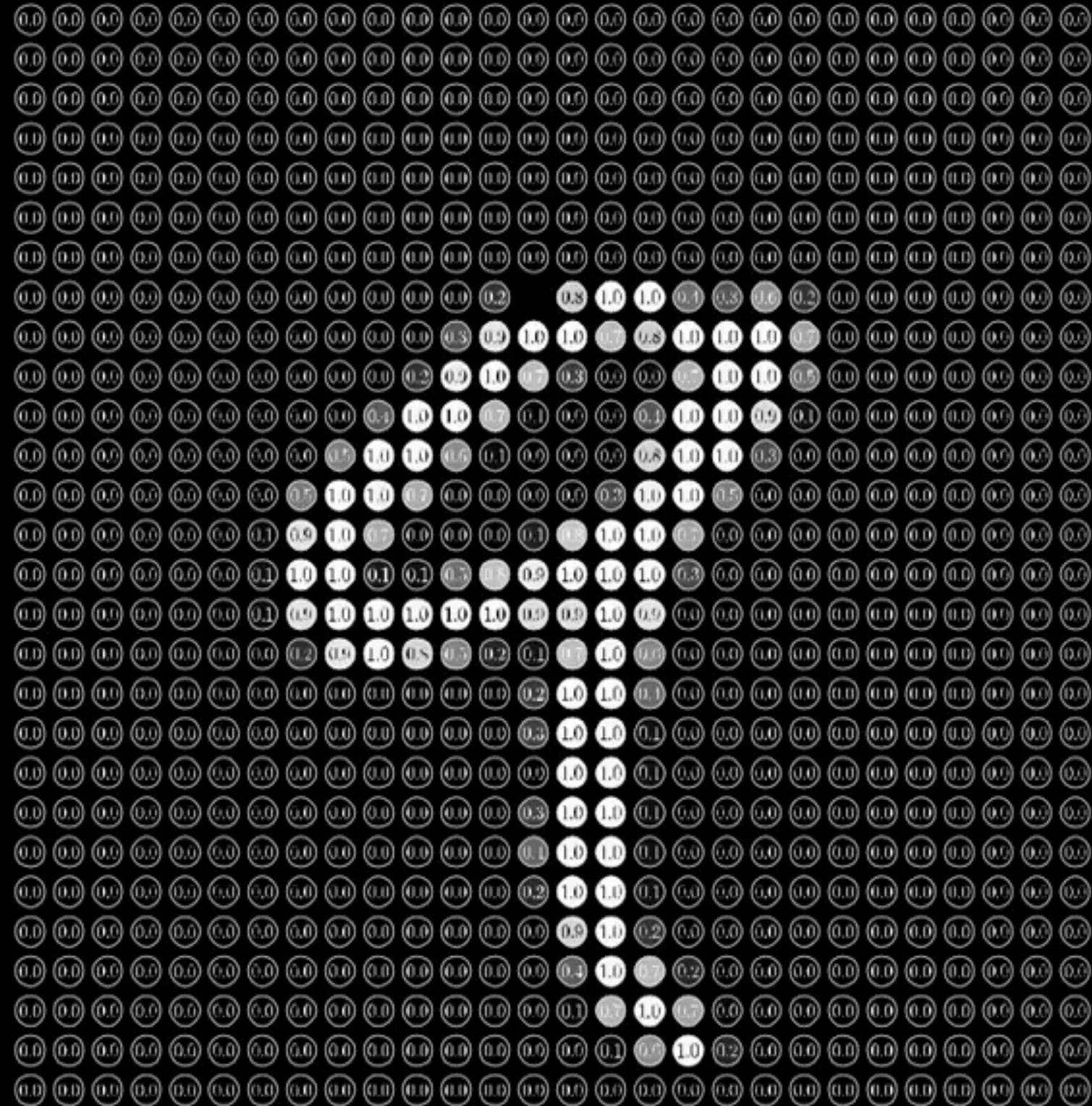


$$28 \times 28 = 784$$

0.02

28

28 {

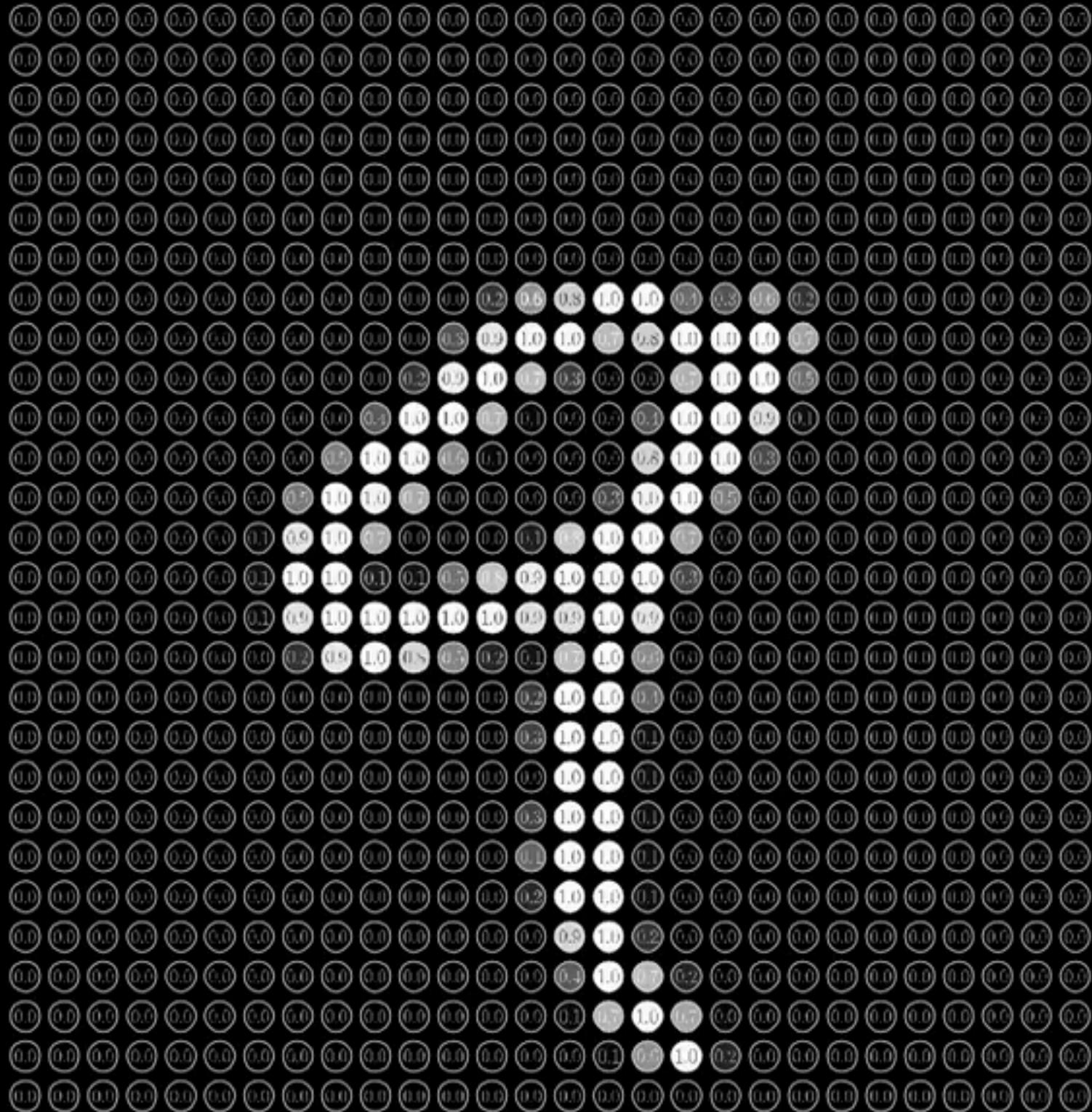


$$28 \times 28 = 784$$

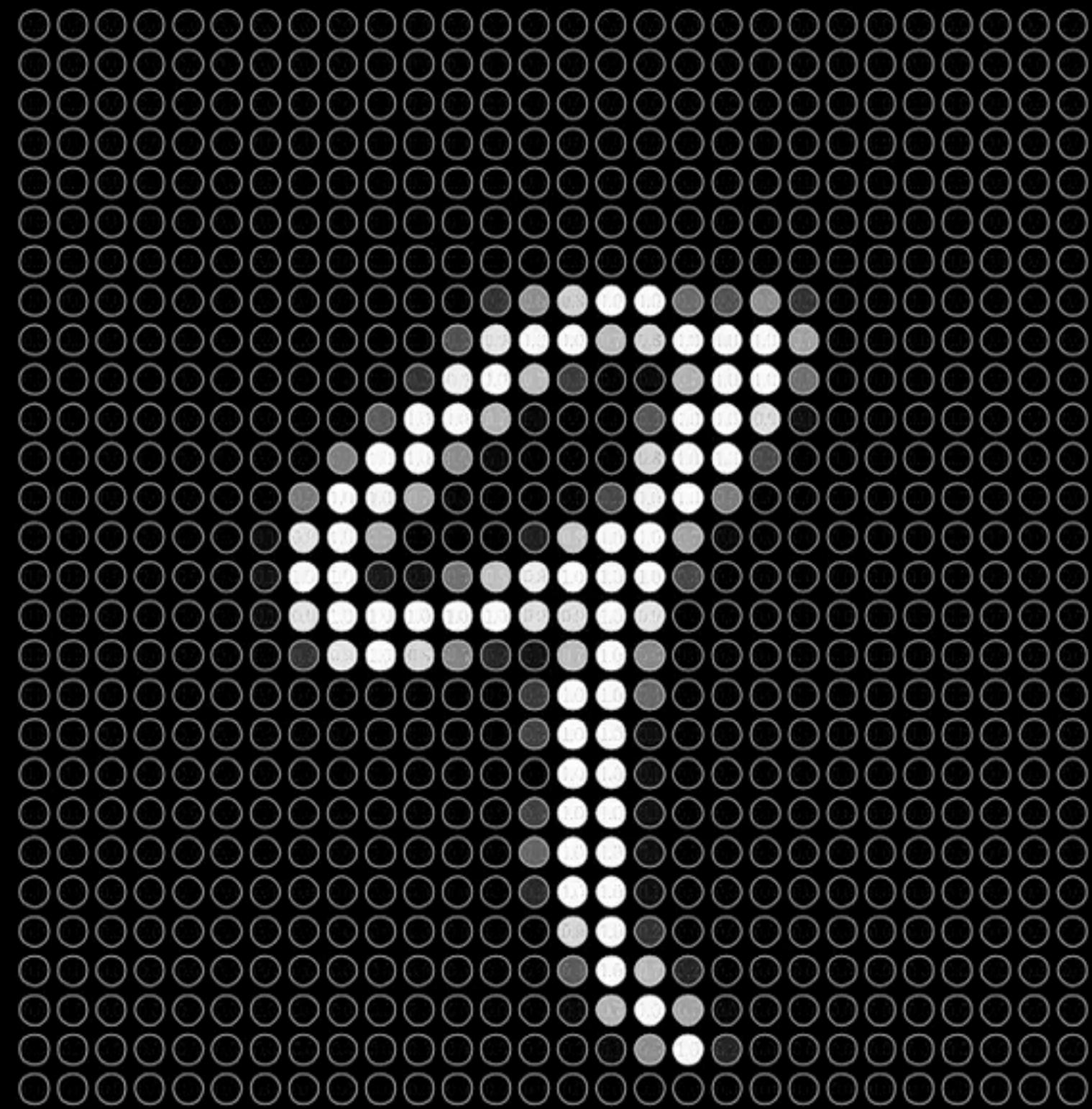
0.34

28

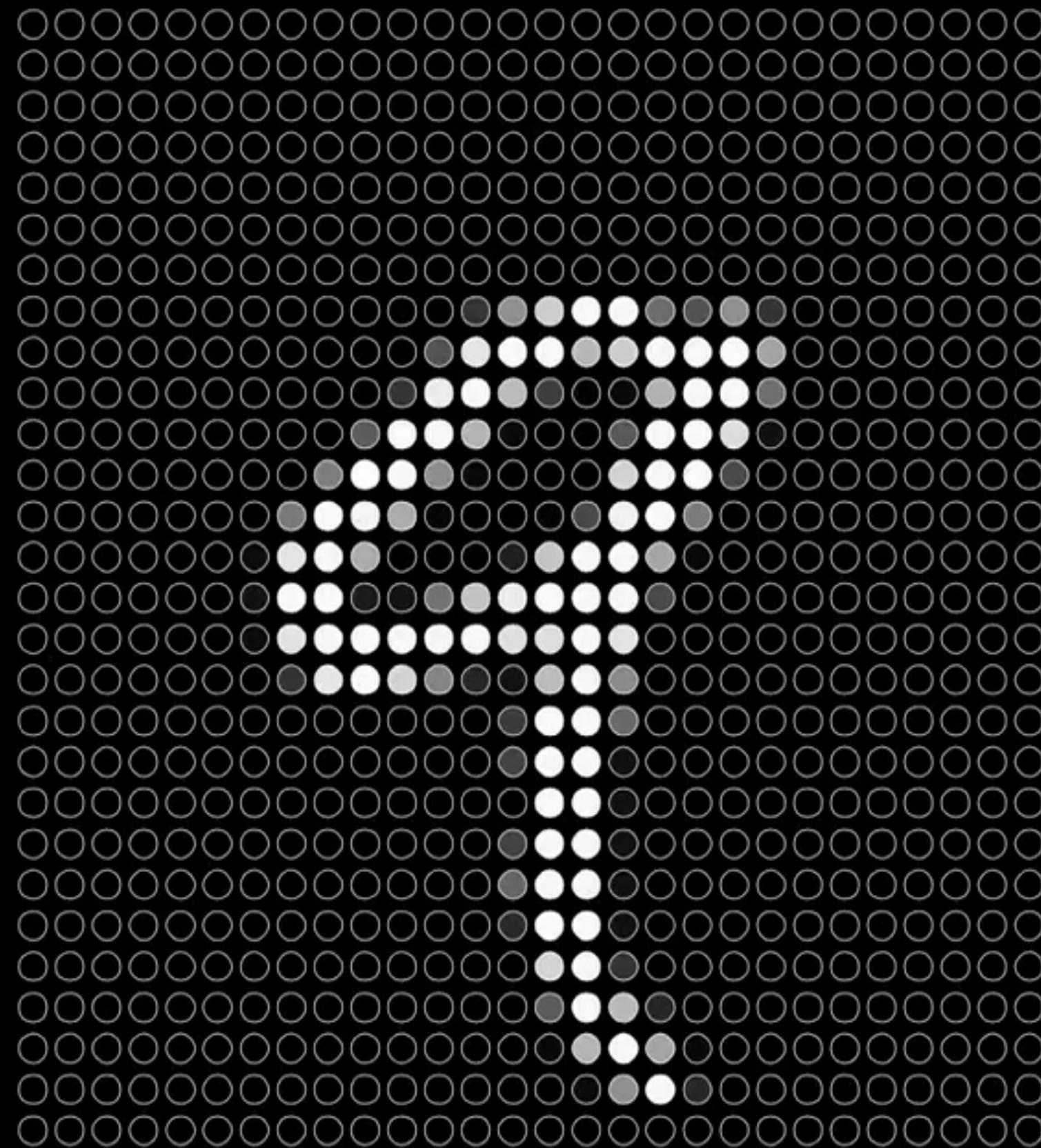
28



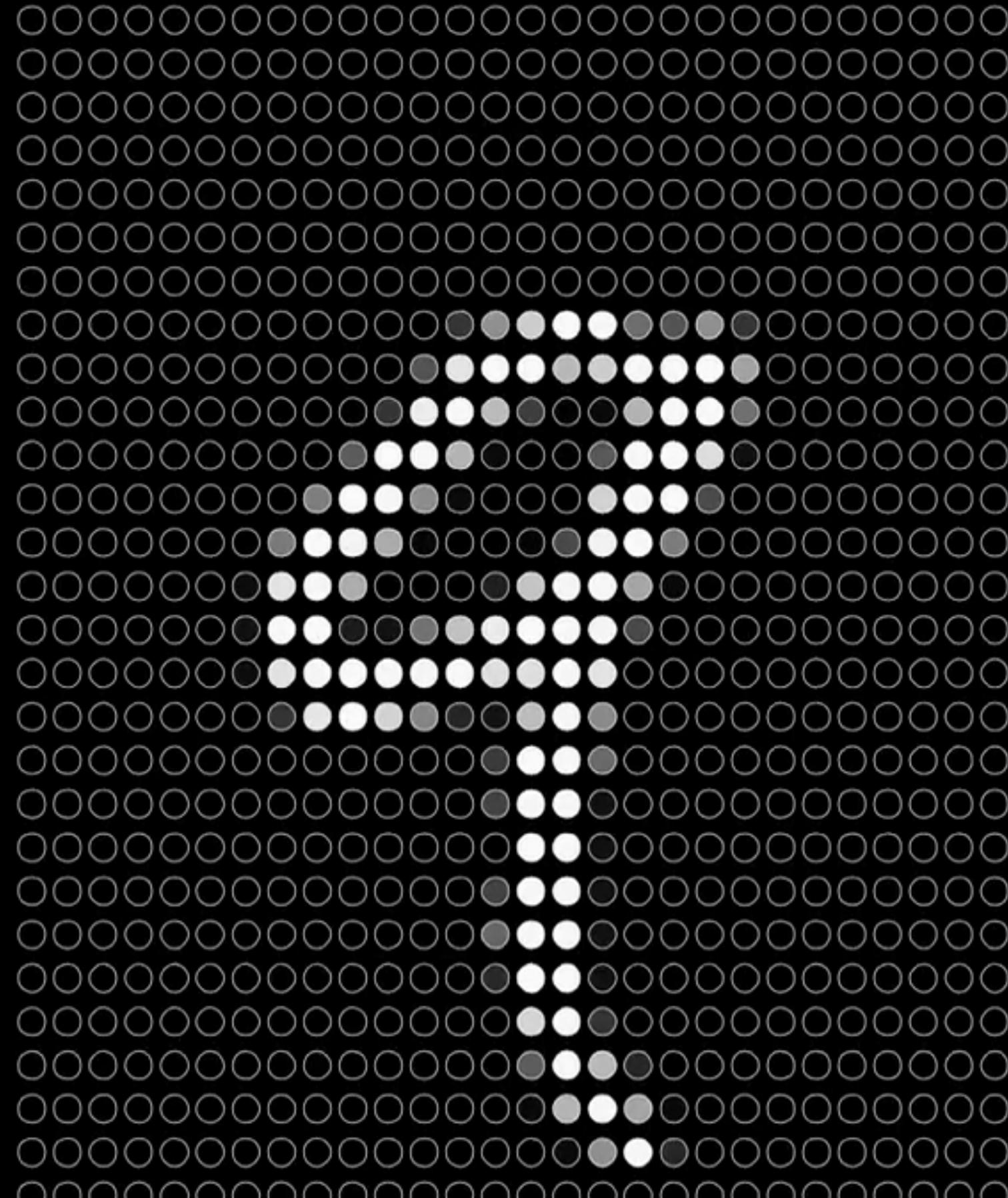
$$28 \times 28 = 784$$



784



784

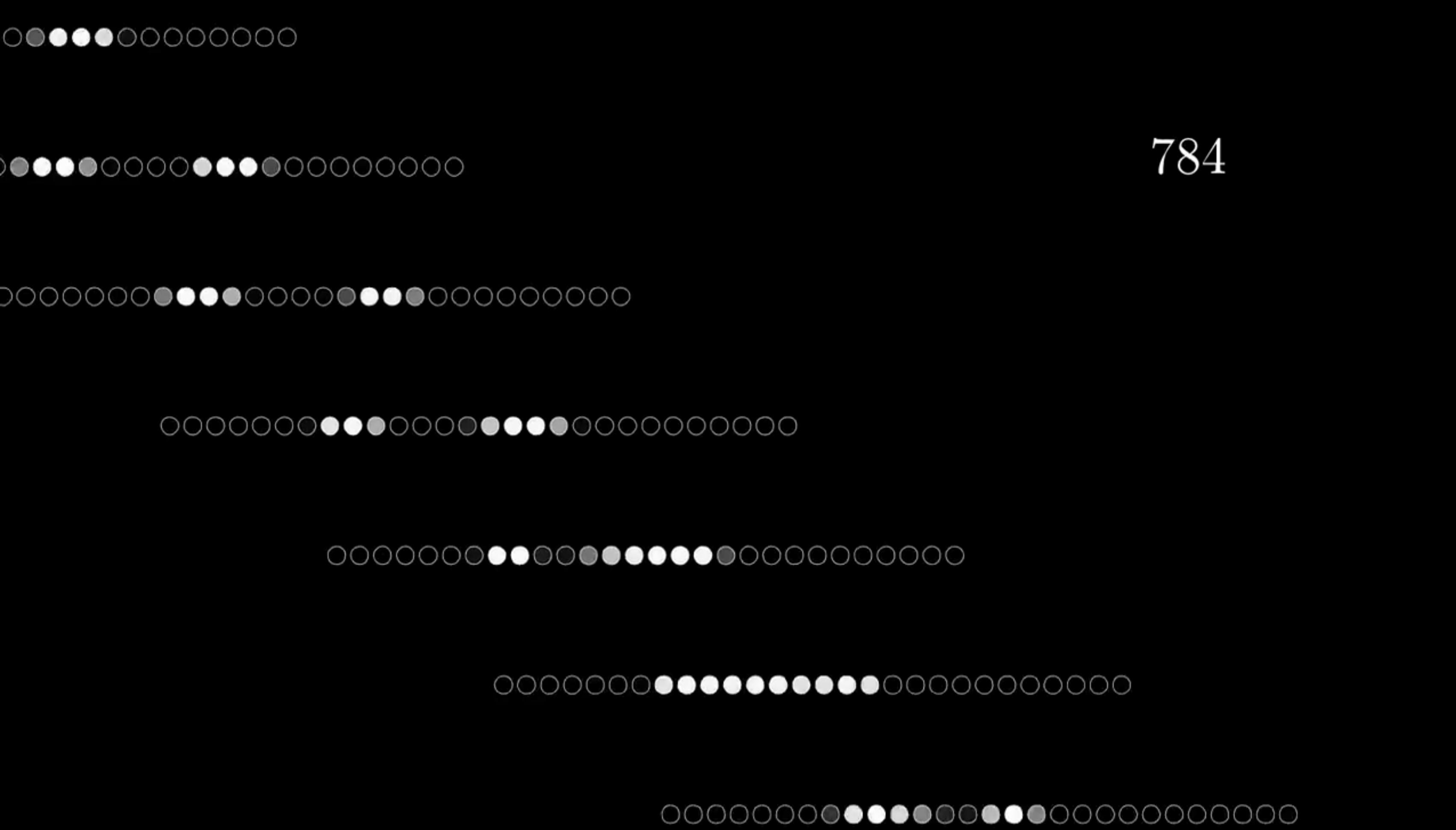


784

A 20x20 grid of circles. The color of each circle is determined by its position relative to the center. The center circle is white. Moving away from the center, the circles transition through shades of gray: light gray for the first few steps outwards, medium gray for the next few, and dark gray for the outermost layers. The pattern repeats every 4 rows and 4 columns, creating a periodic radial gradient effect.

784

784



784

784

○○○○○

○○○○○○

784



784

○○○○○○○○ ··· ○○○○○○○○



784

○○○○○○○○ … ○○○○○○○○

784



784

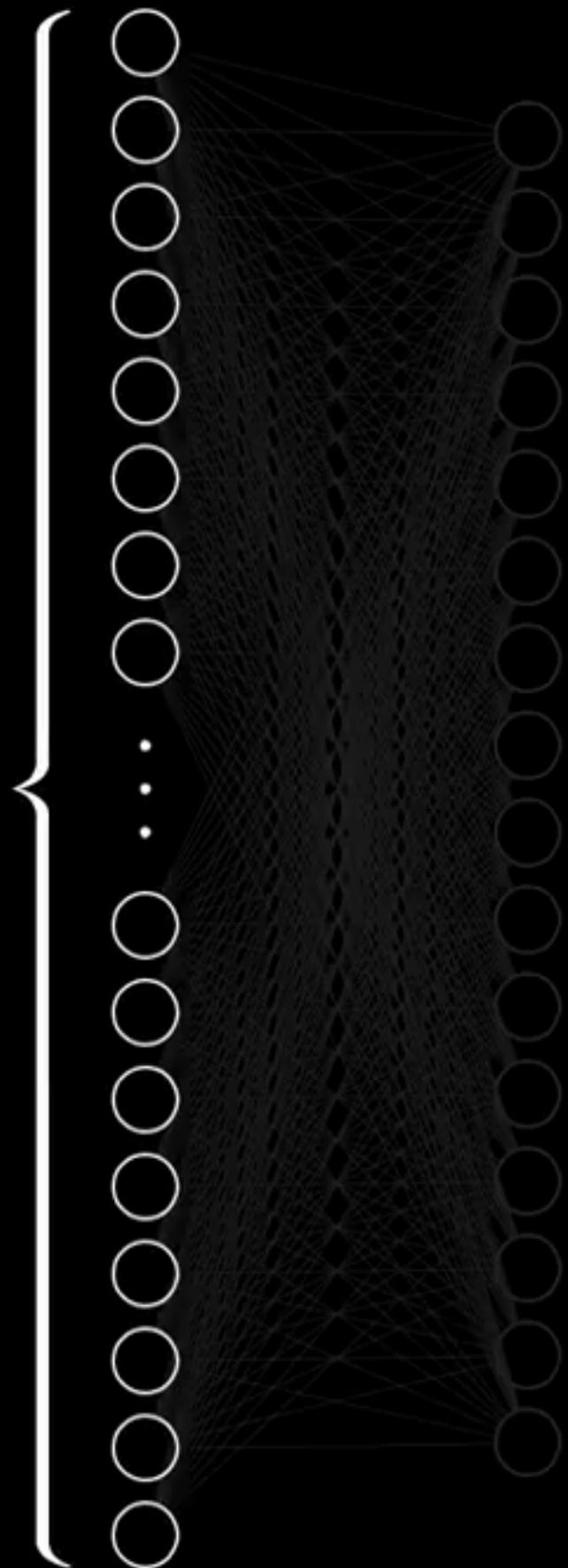


784



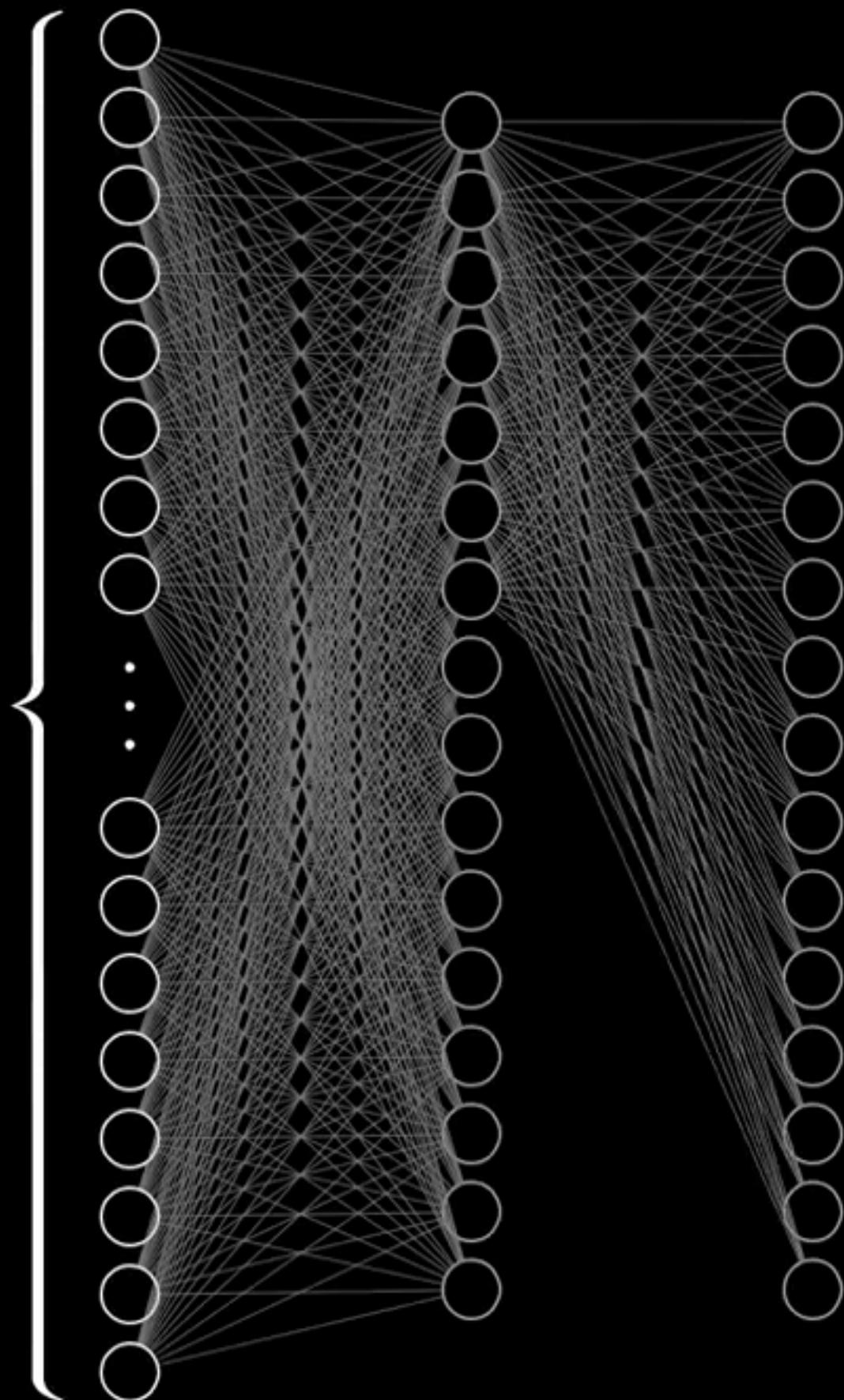


784



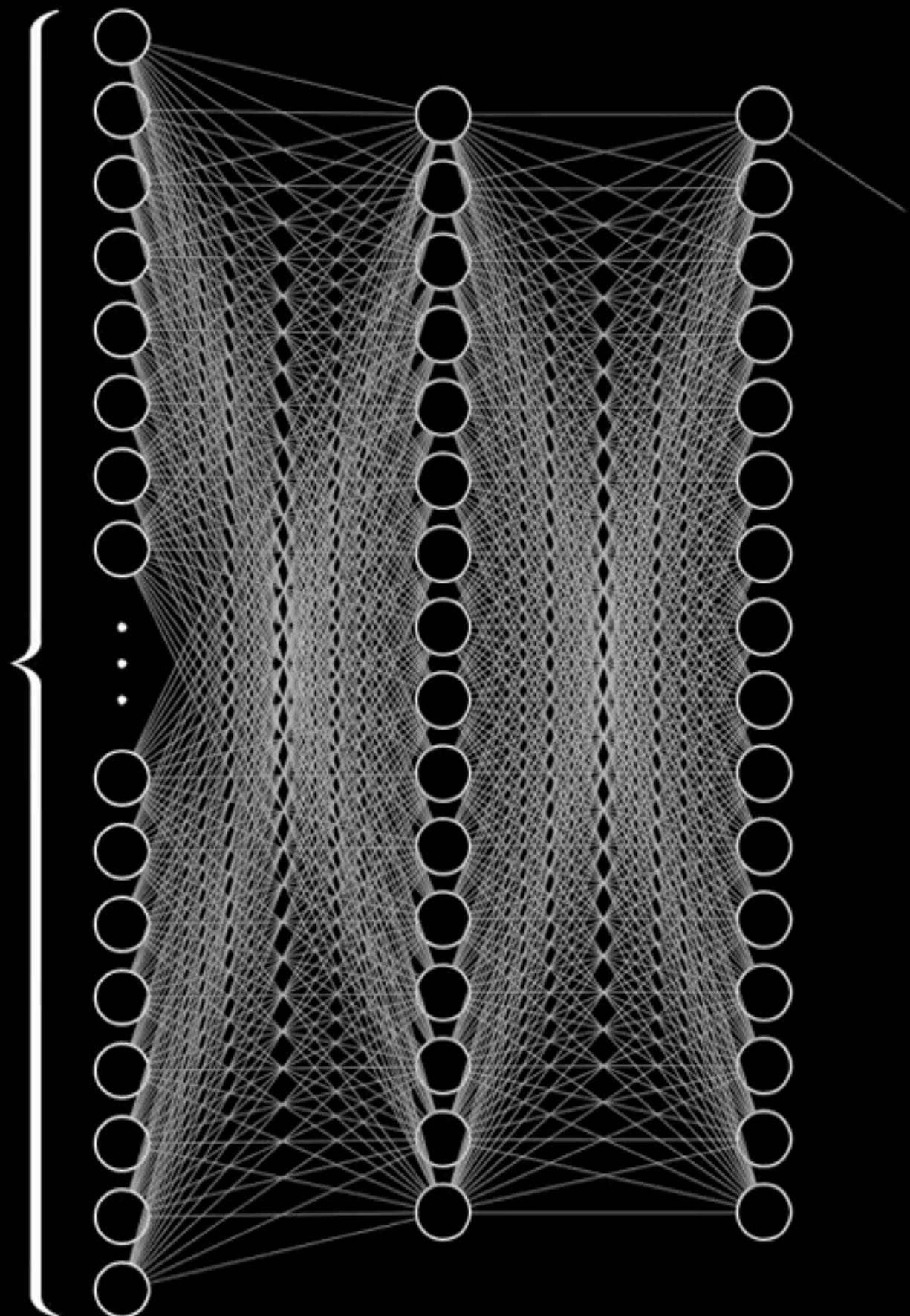


784



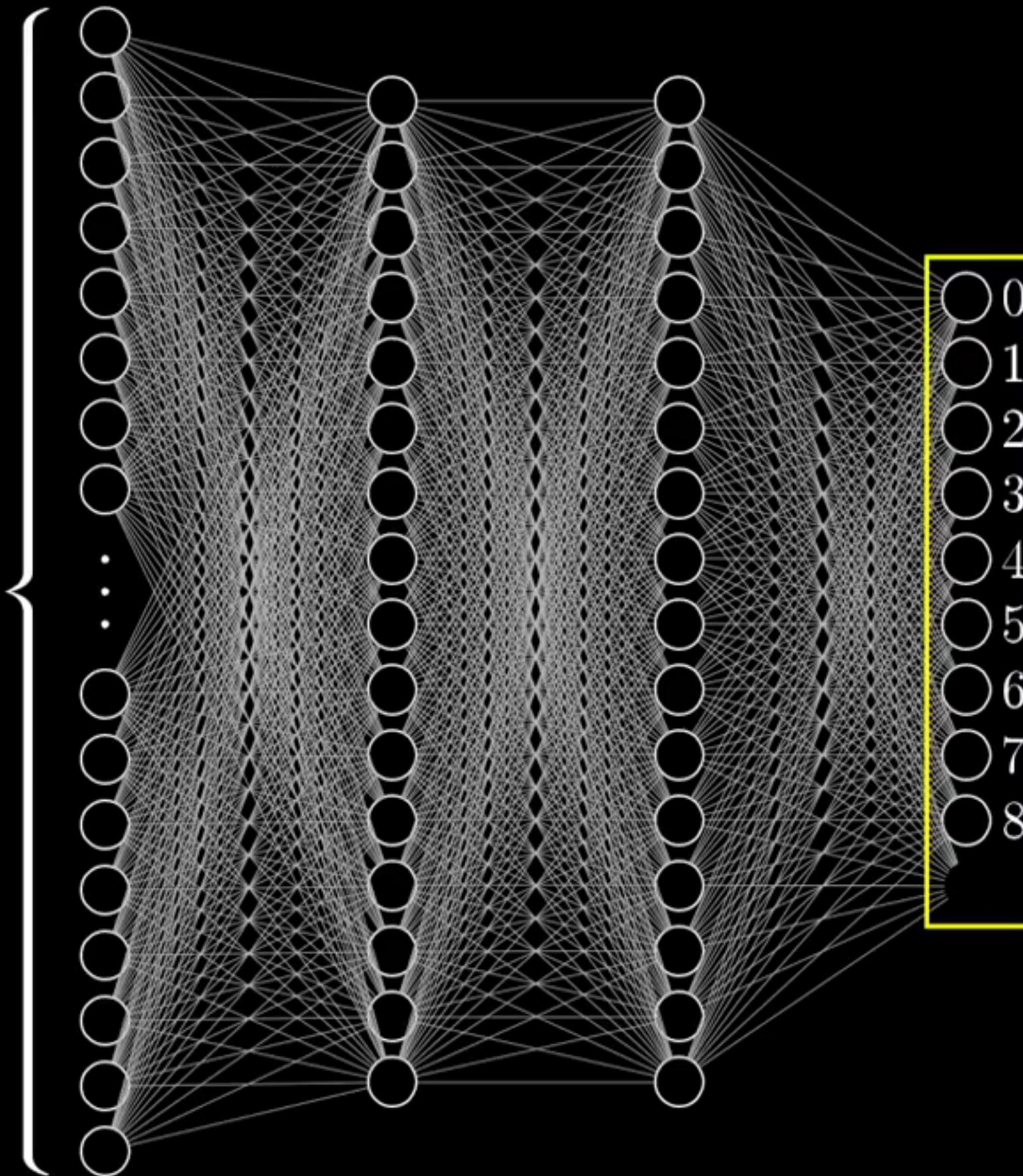


784





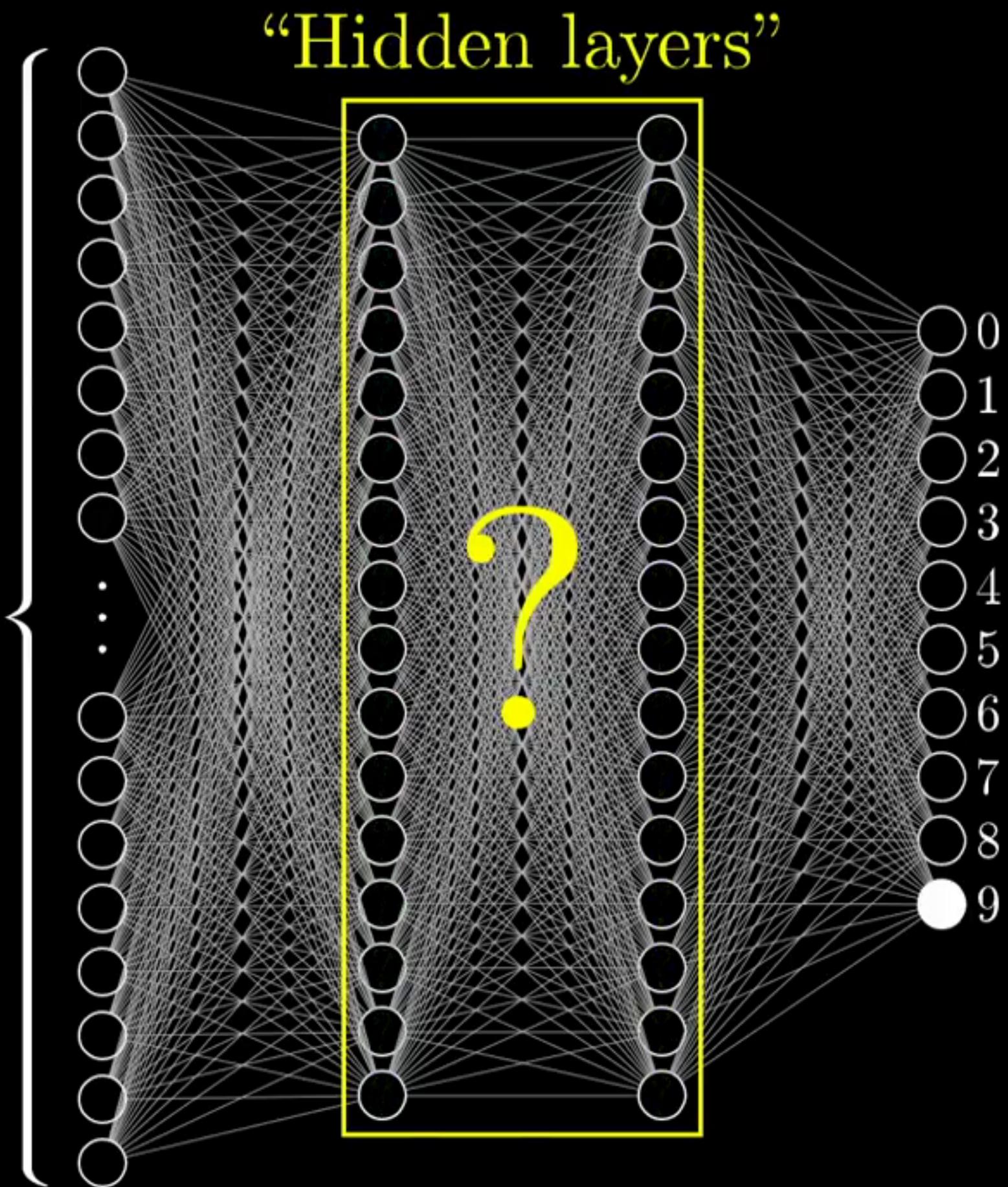
784



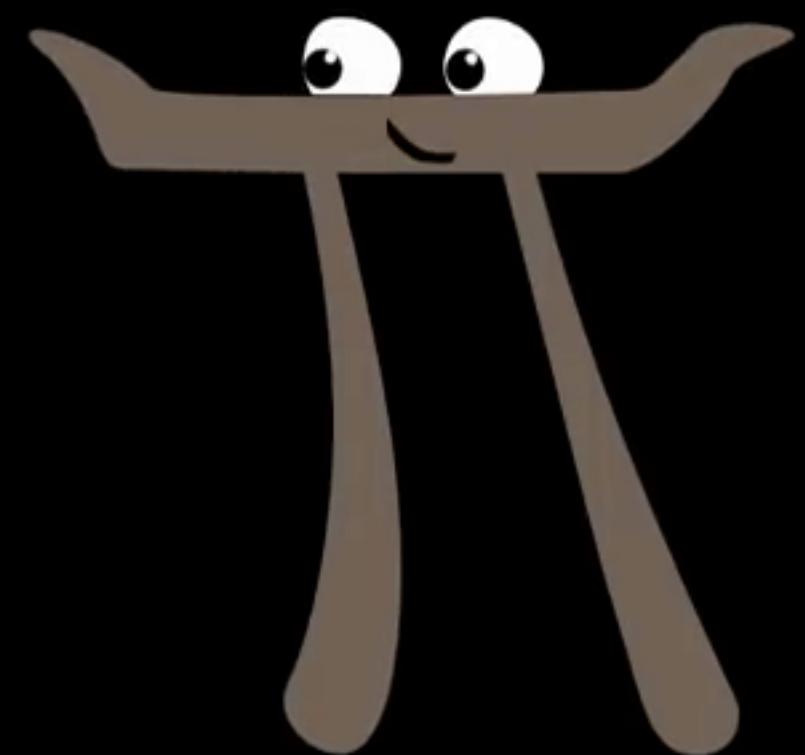
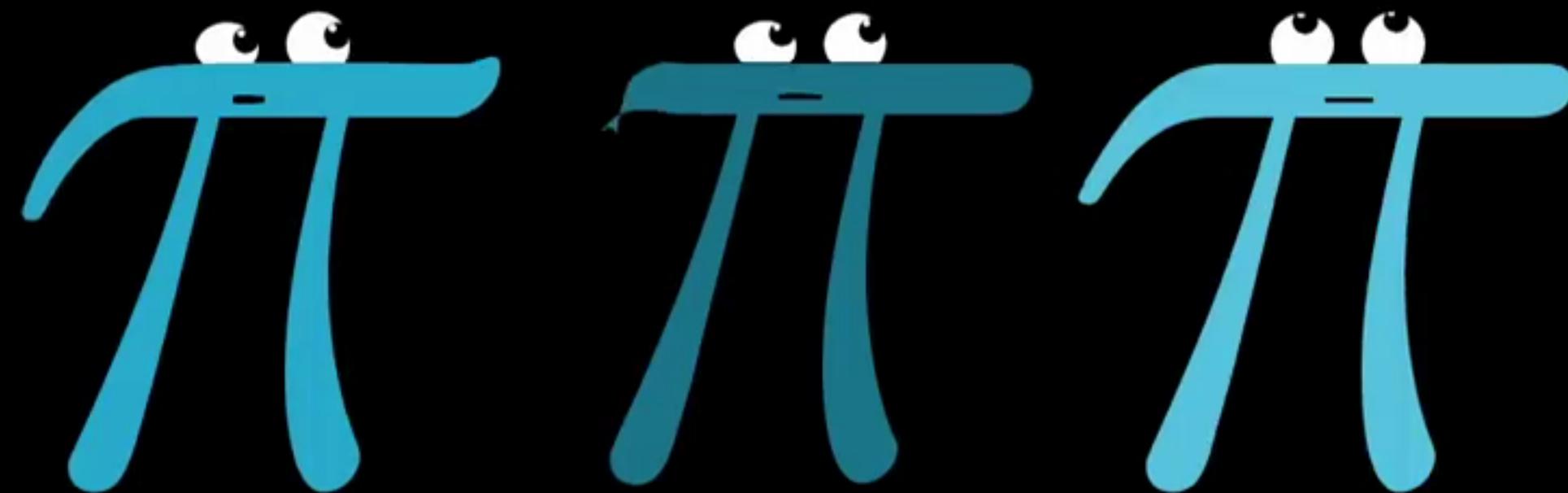
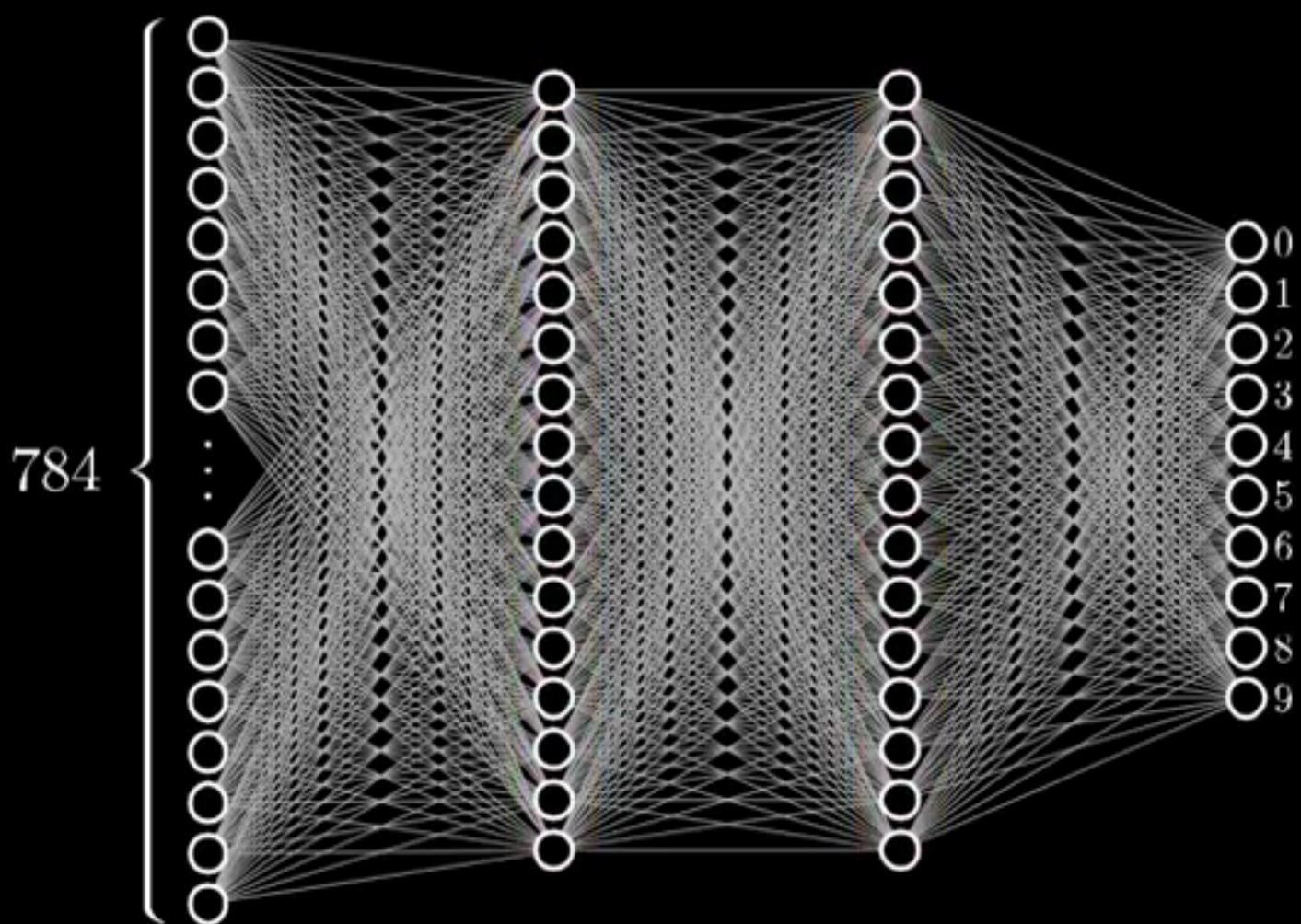
0.97 9



784



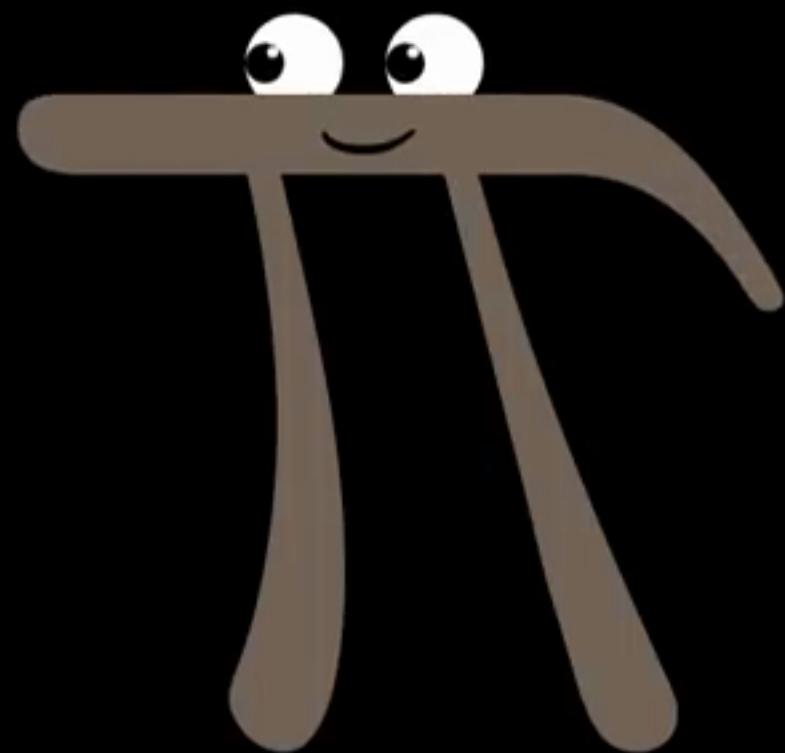
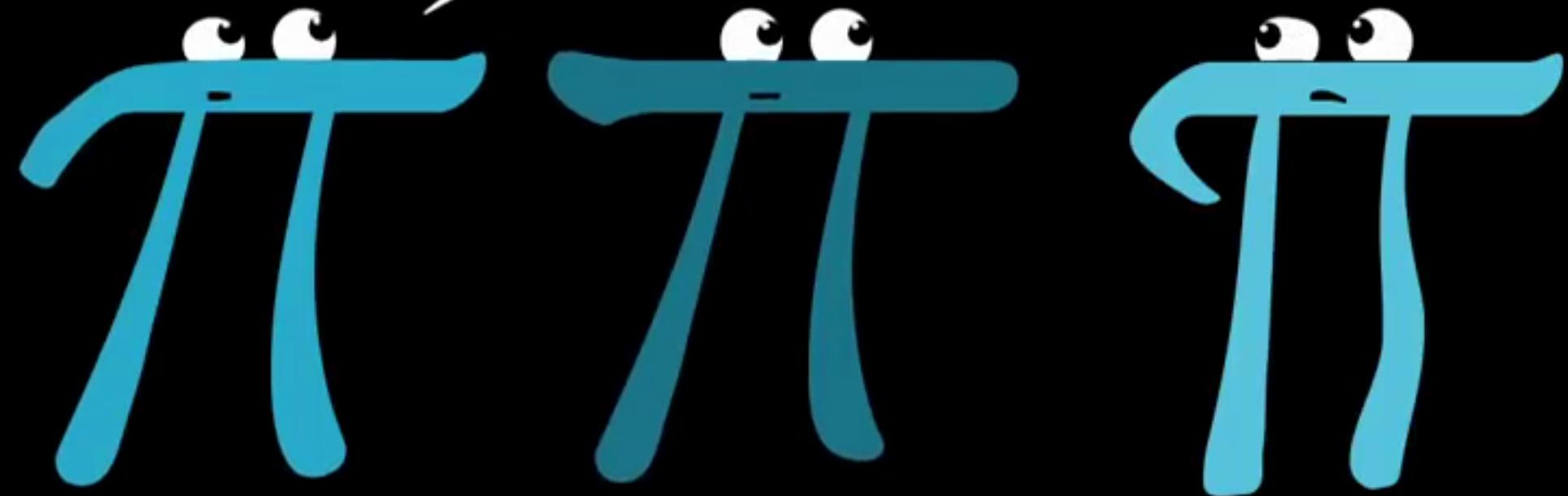
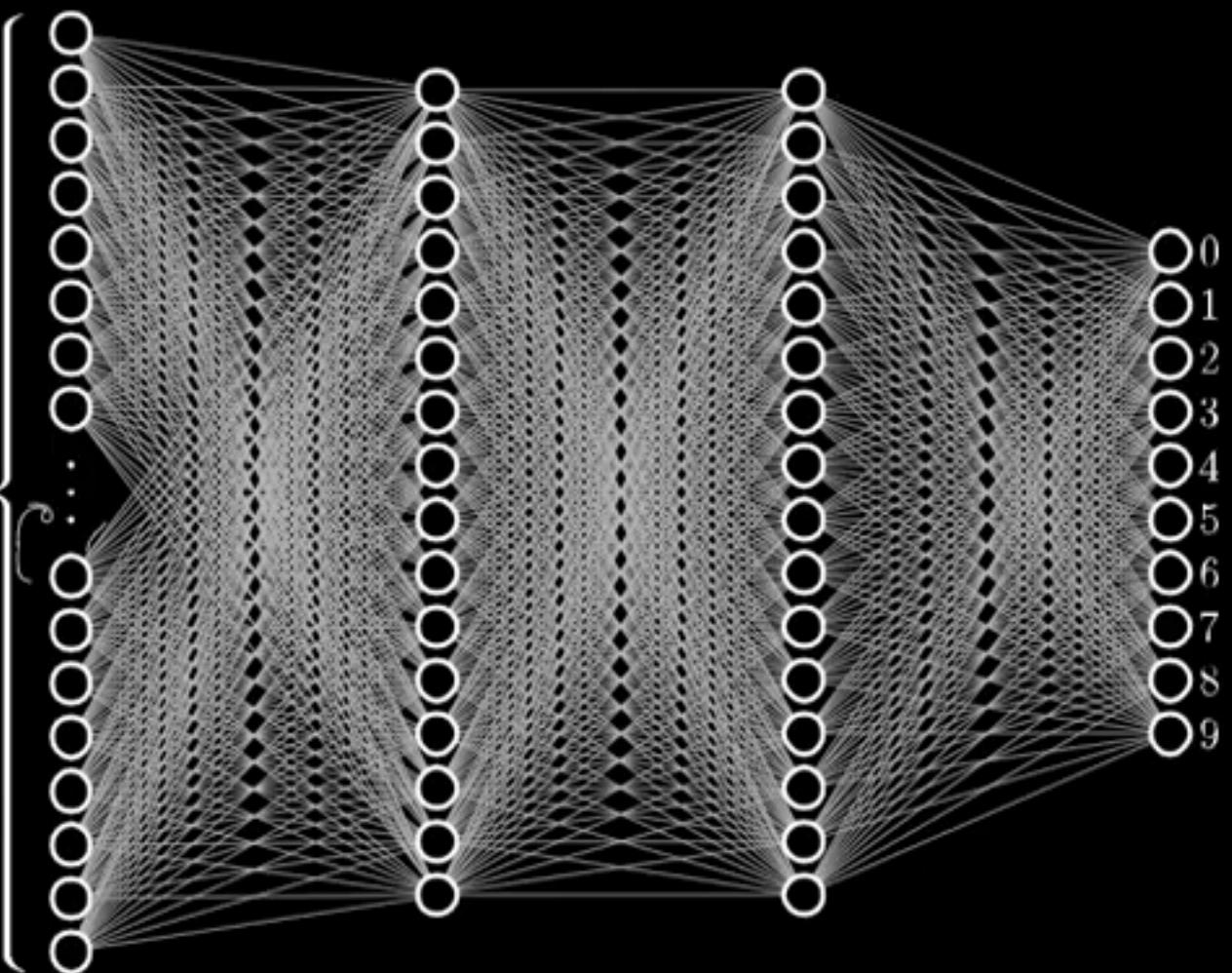
16 neurons each



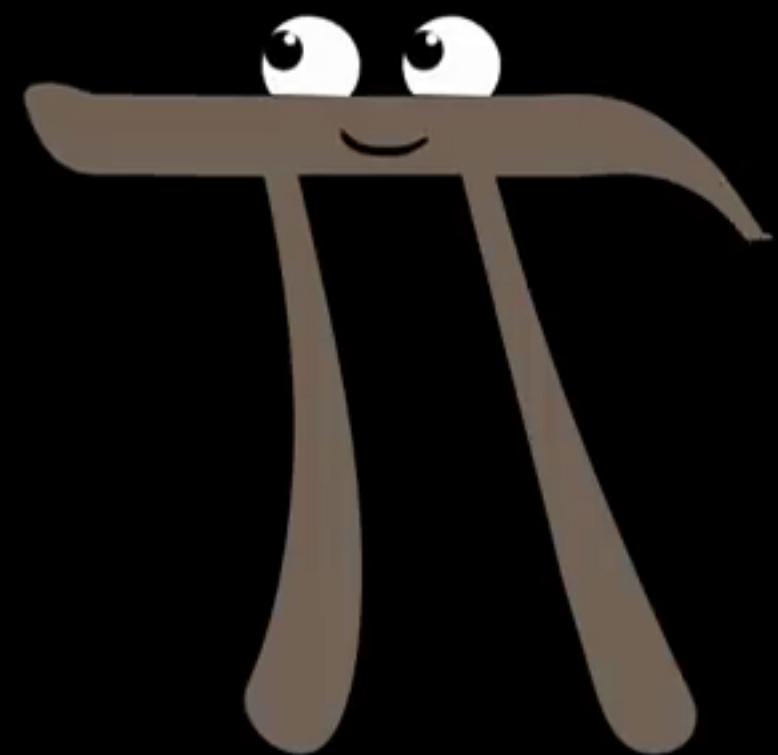
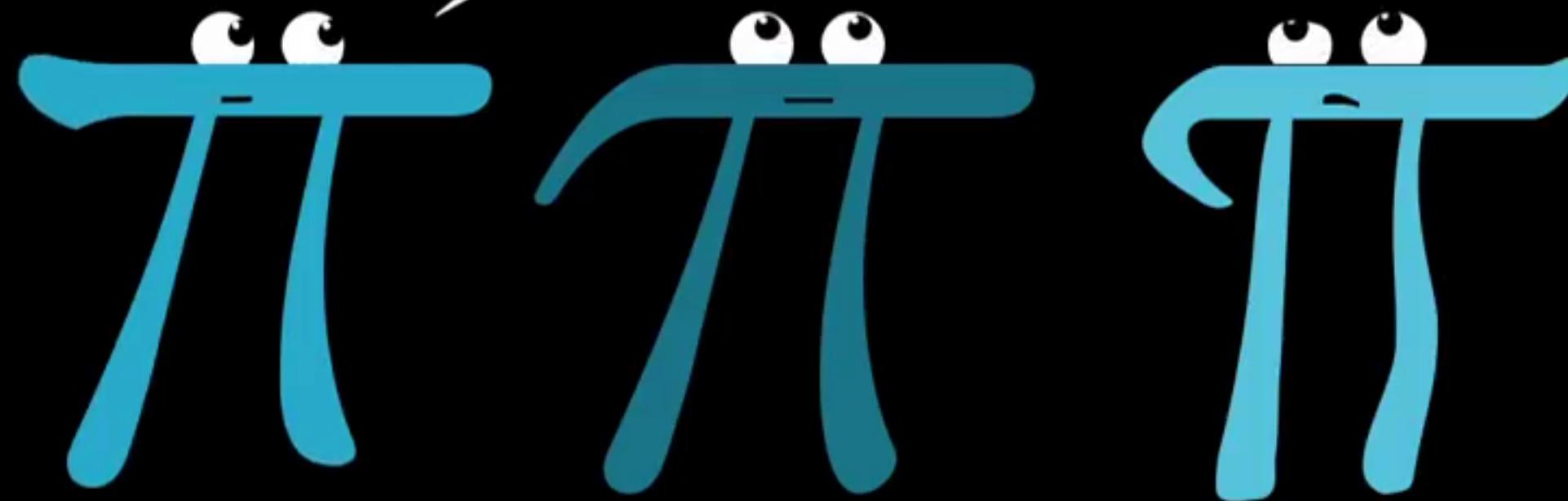
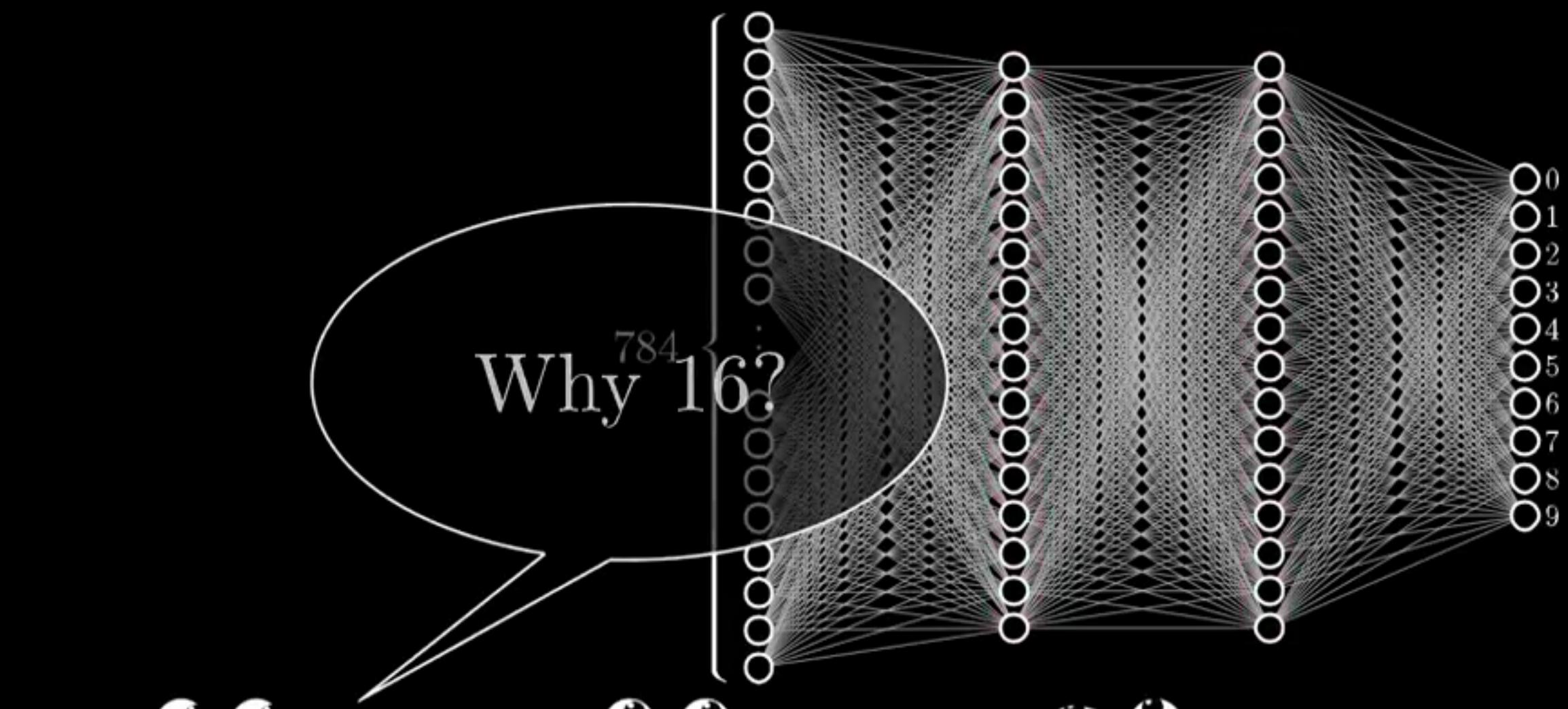
16 neurons each

Why 2
layers?

Why⁷⁸⁴
Why¹

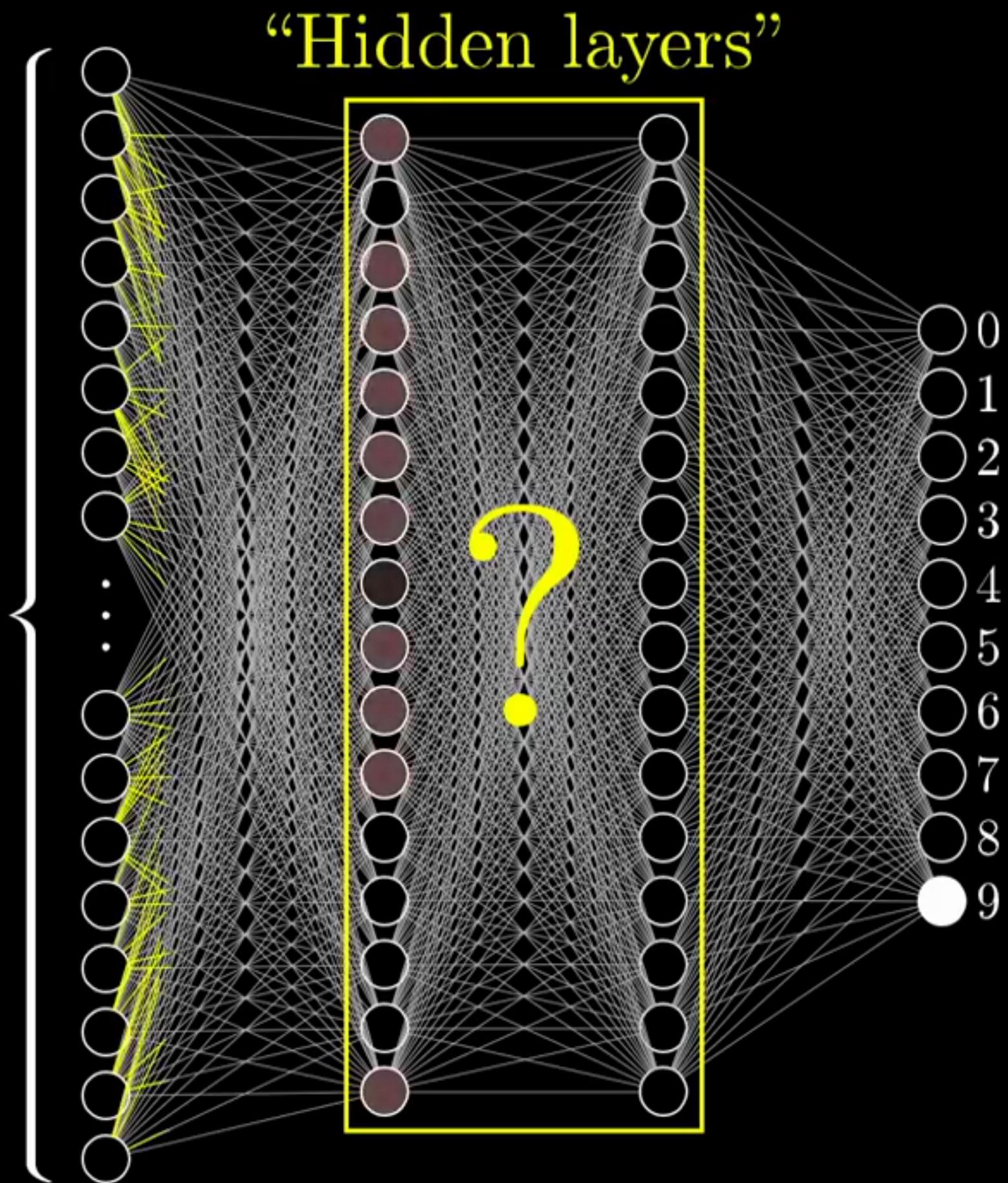


16 neurons each



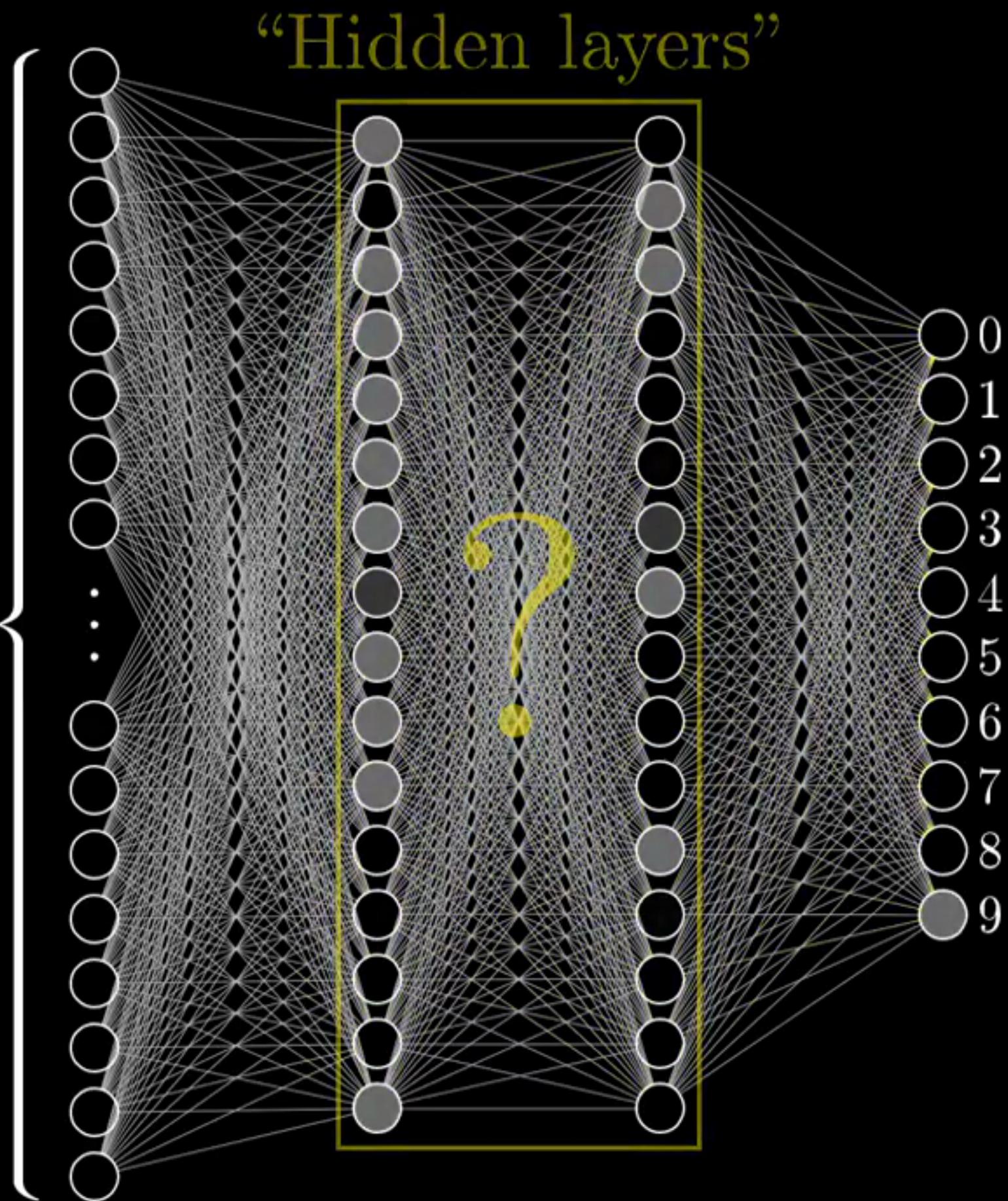


784



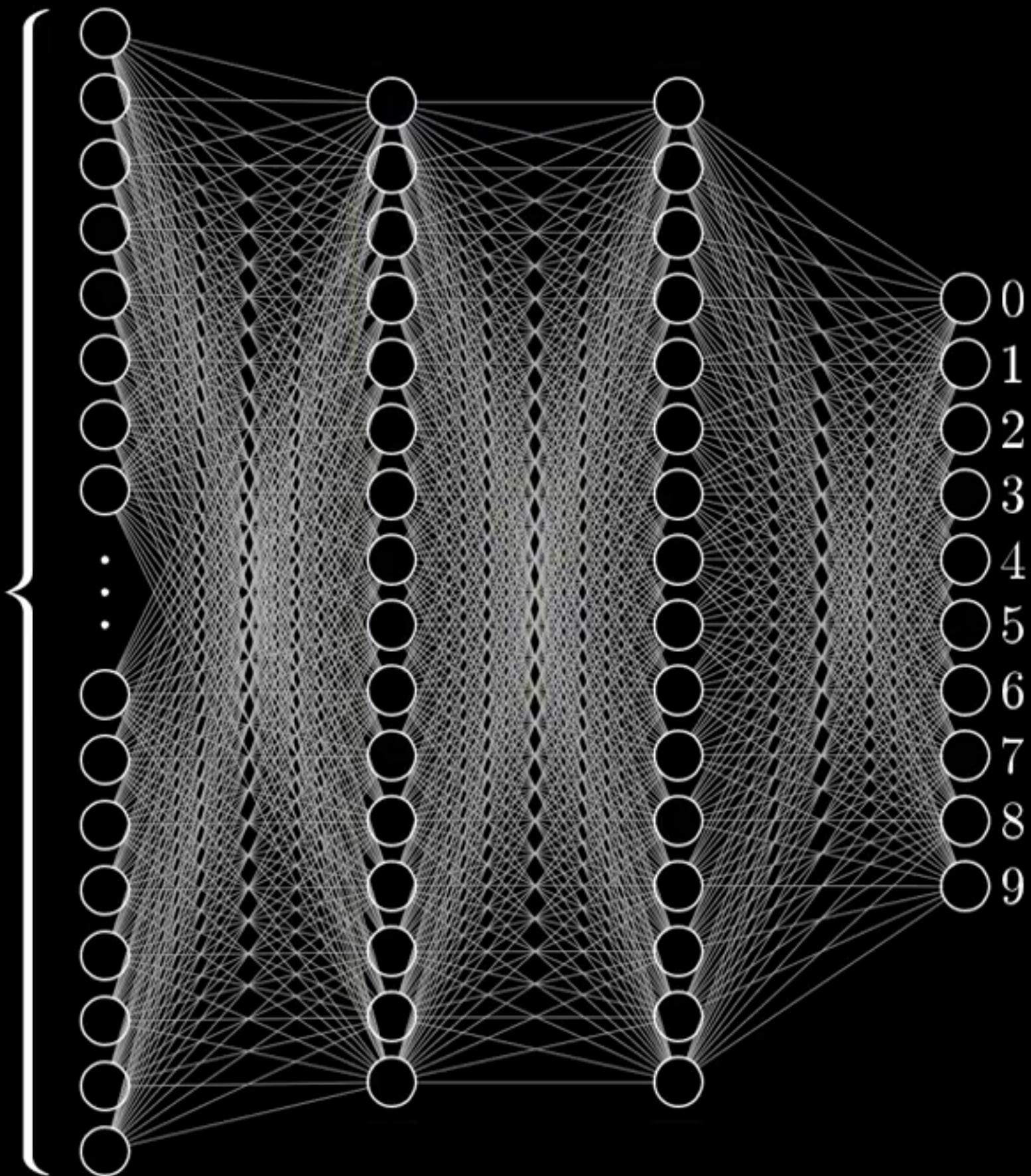


784

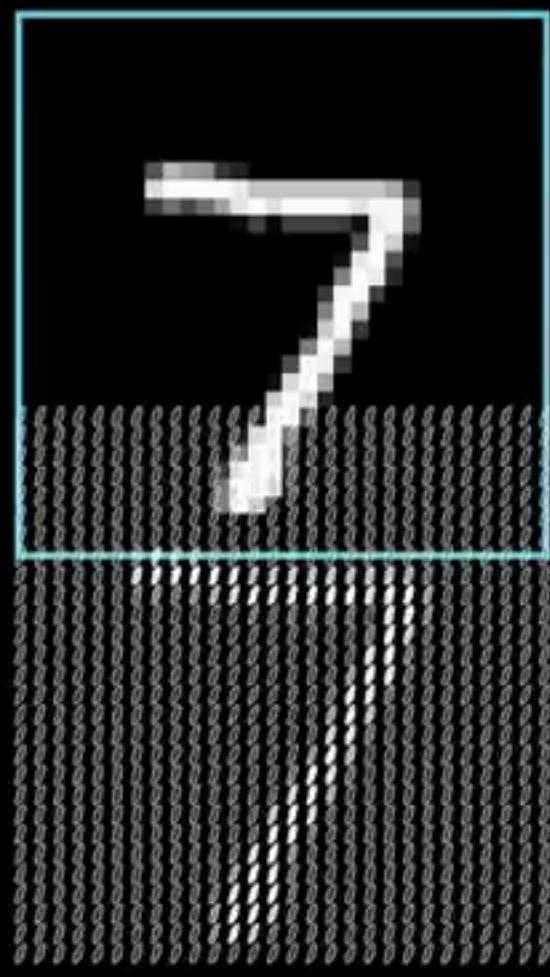




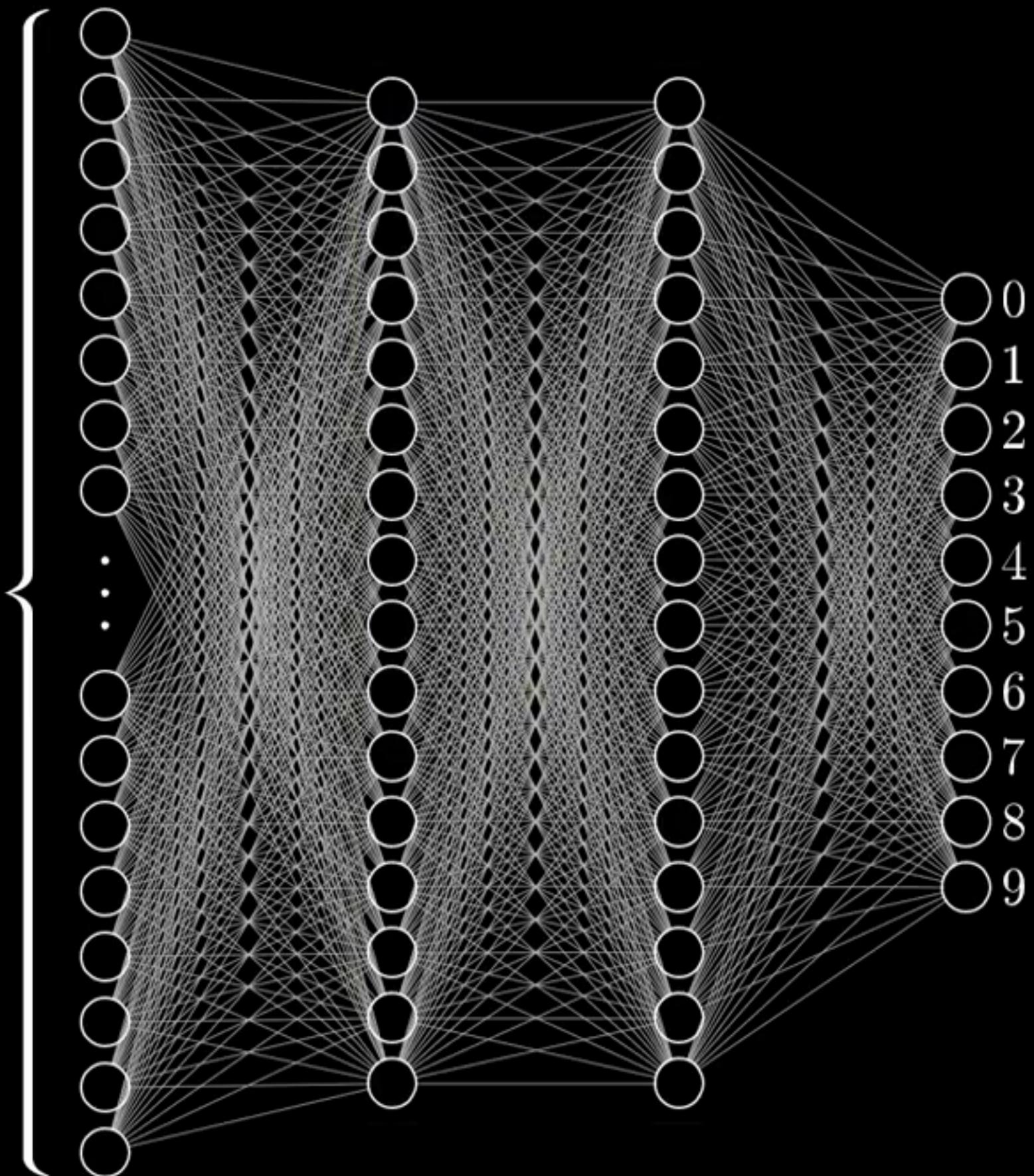
784



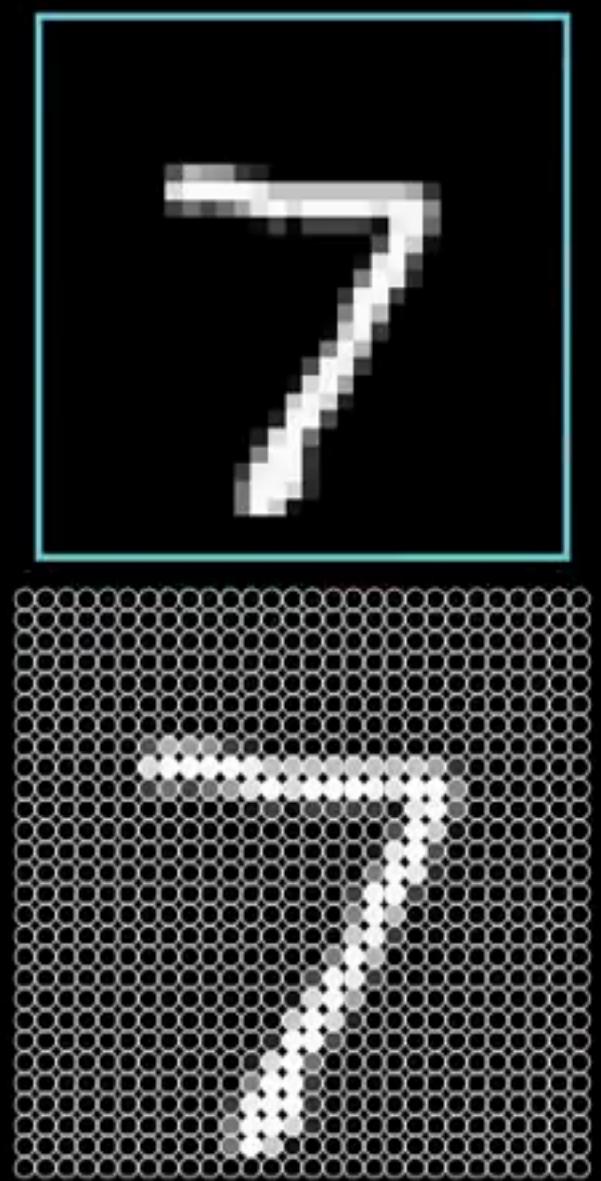
0
1
2
3
4
5
6
7
8
9



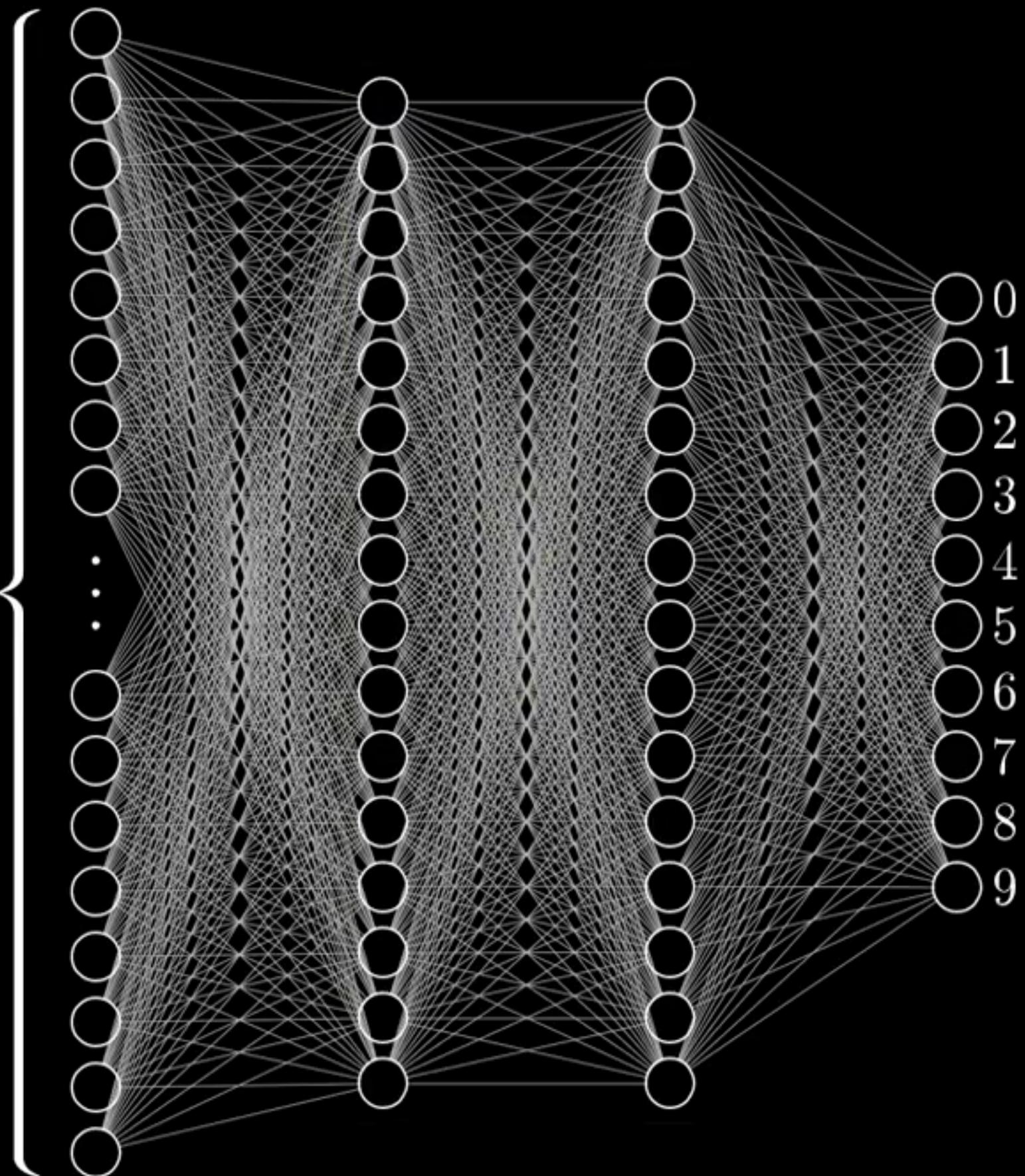
784

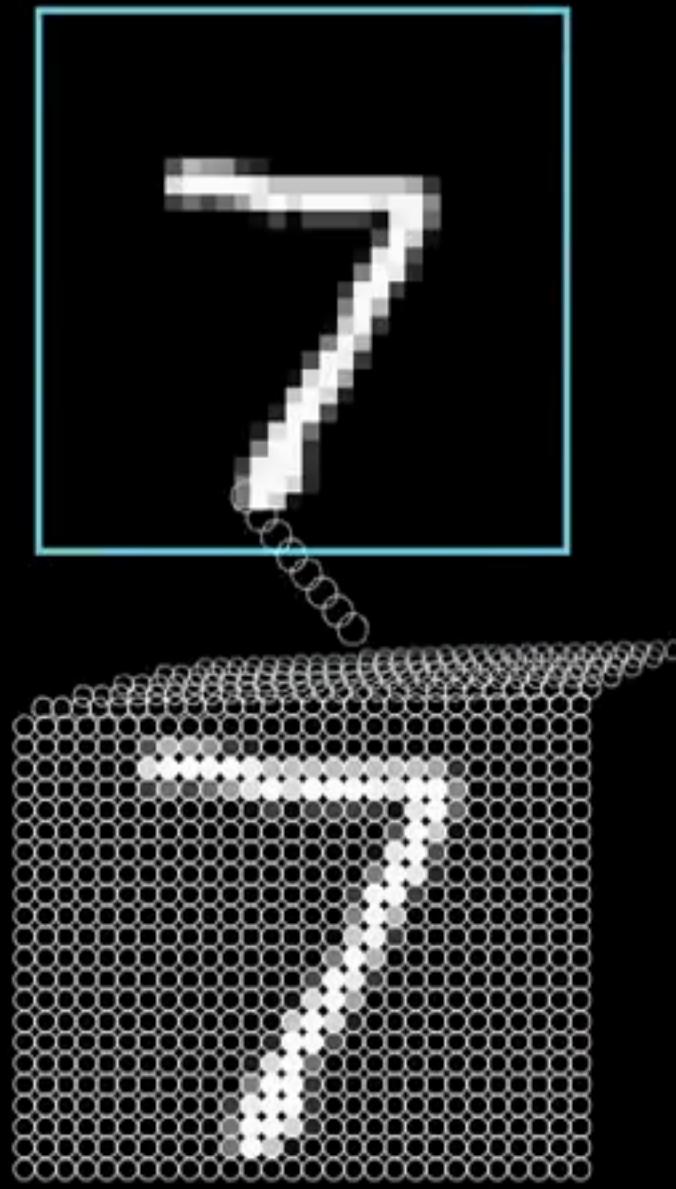


- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

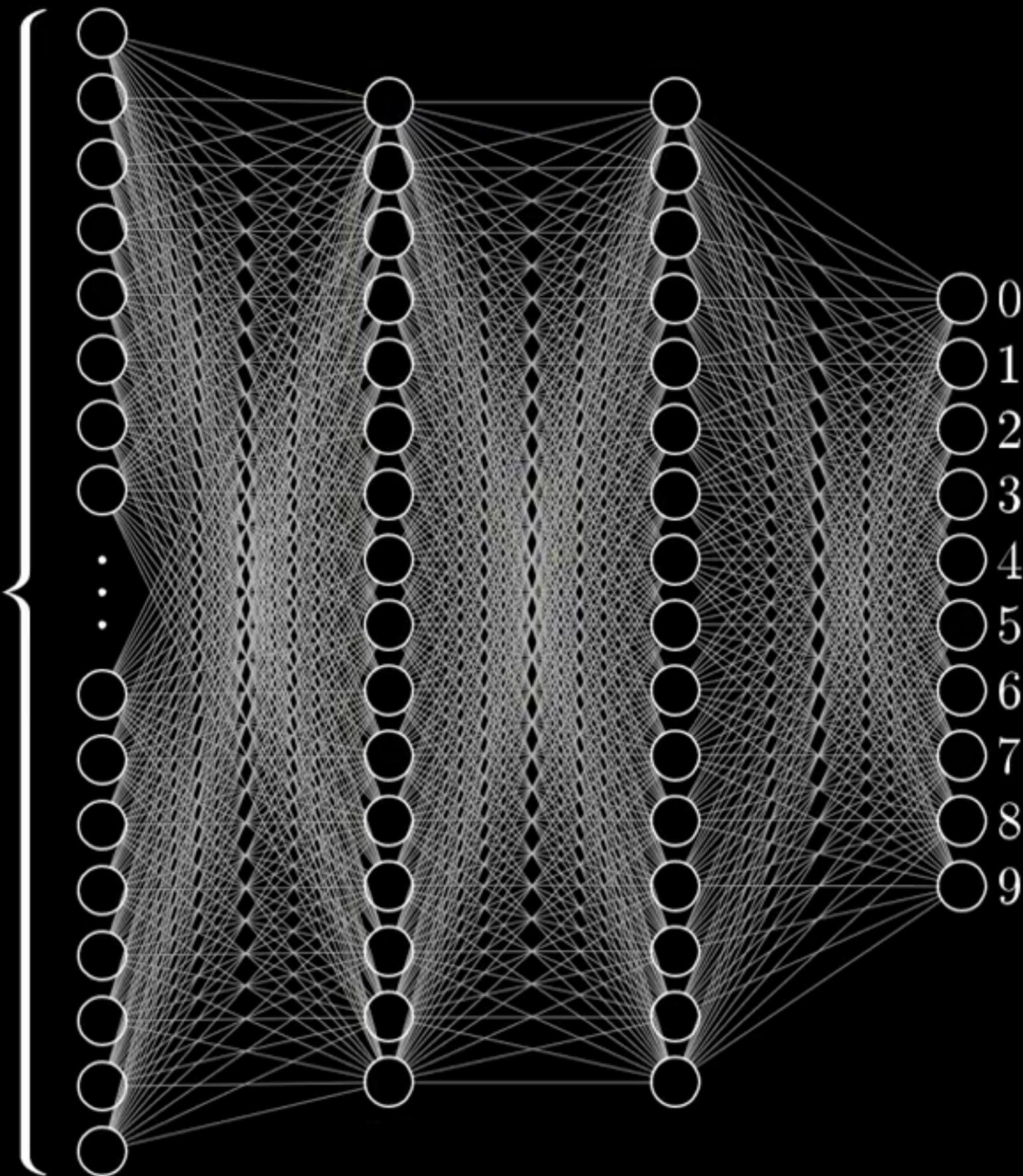


784



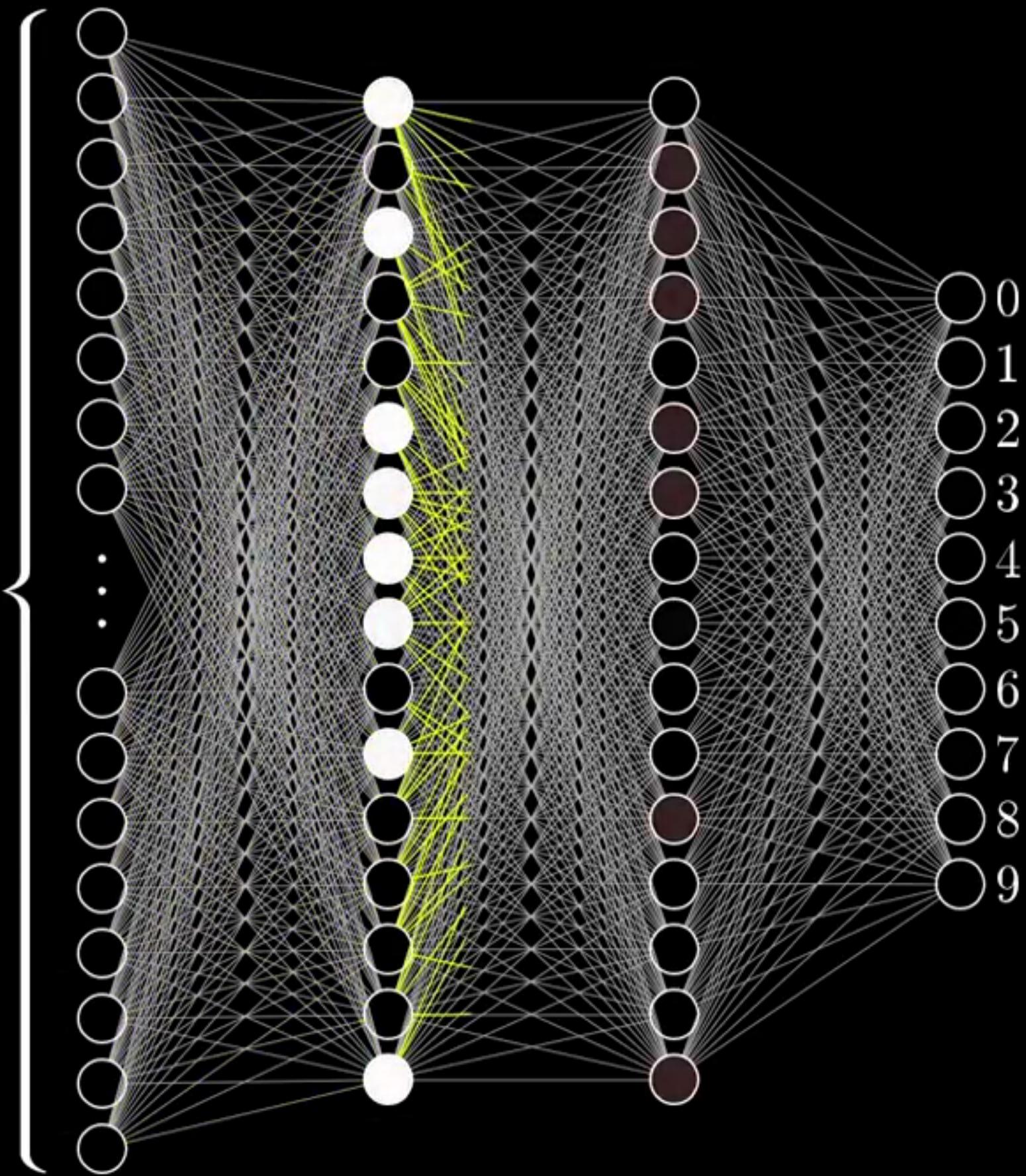


784



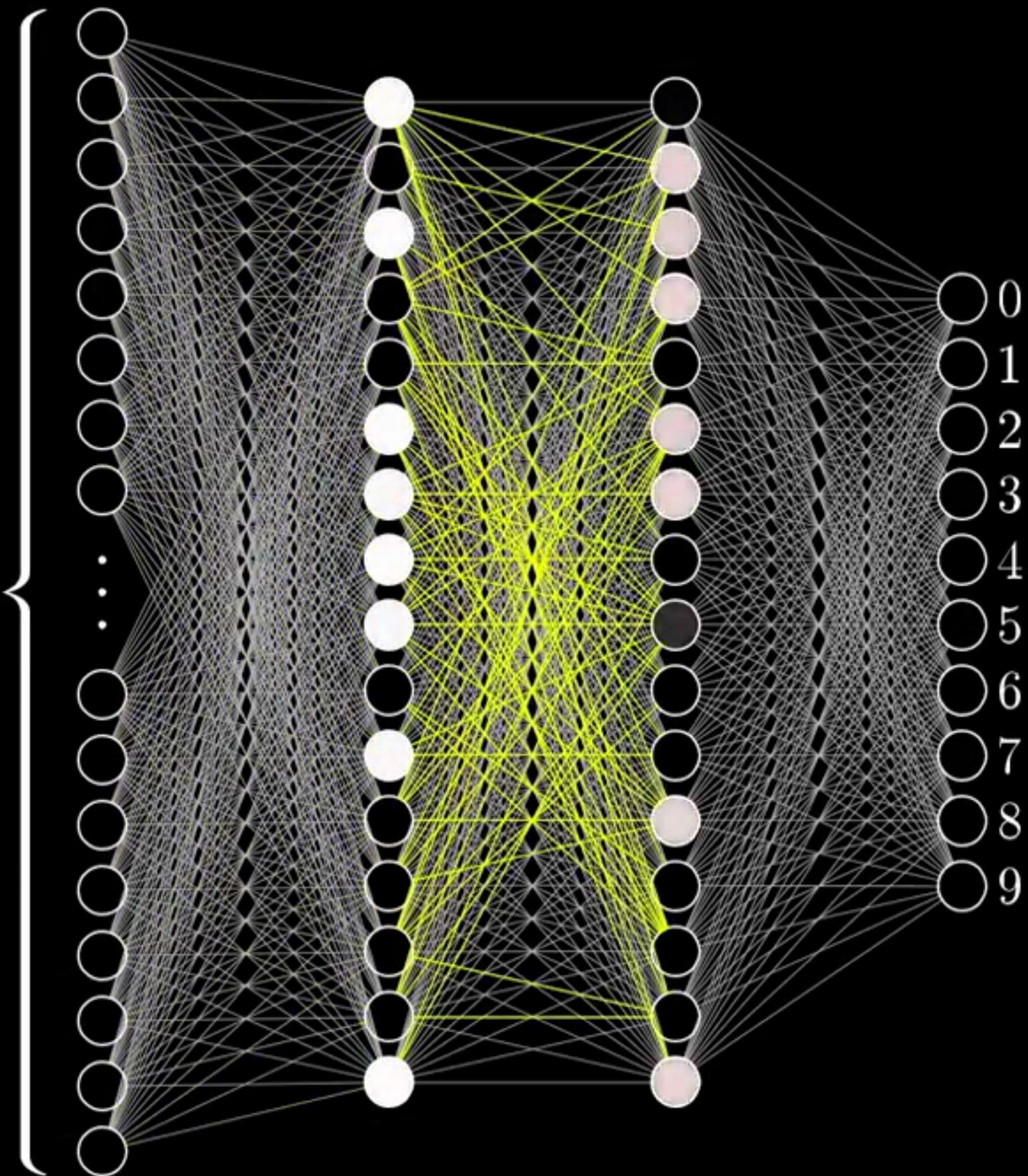


784



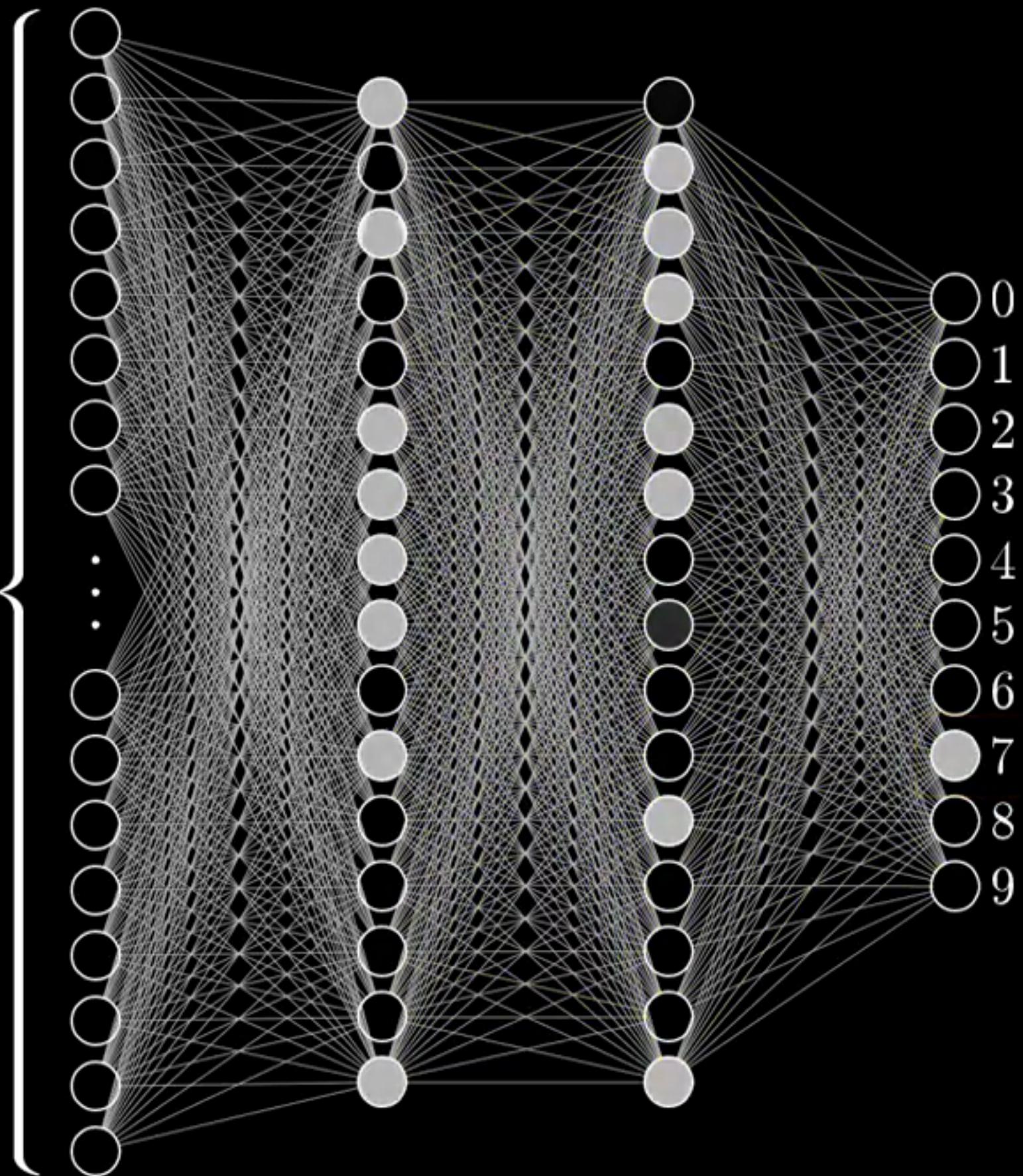


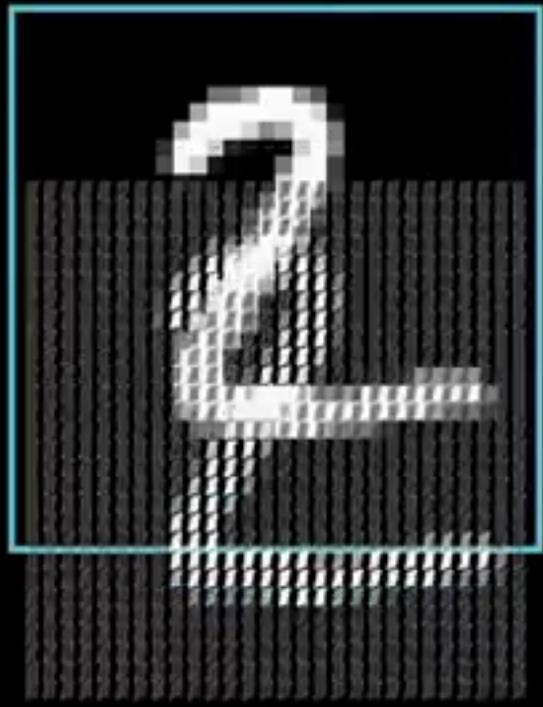
784



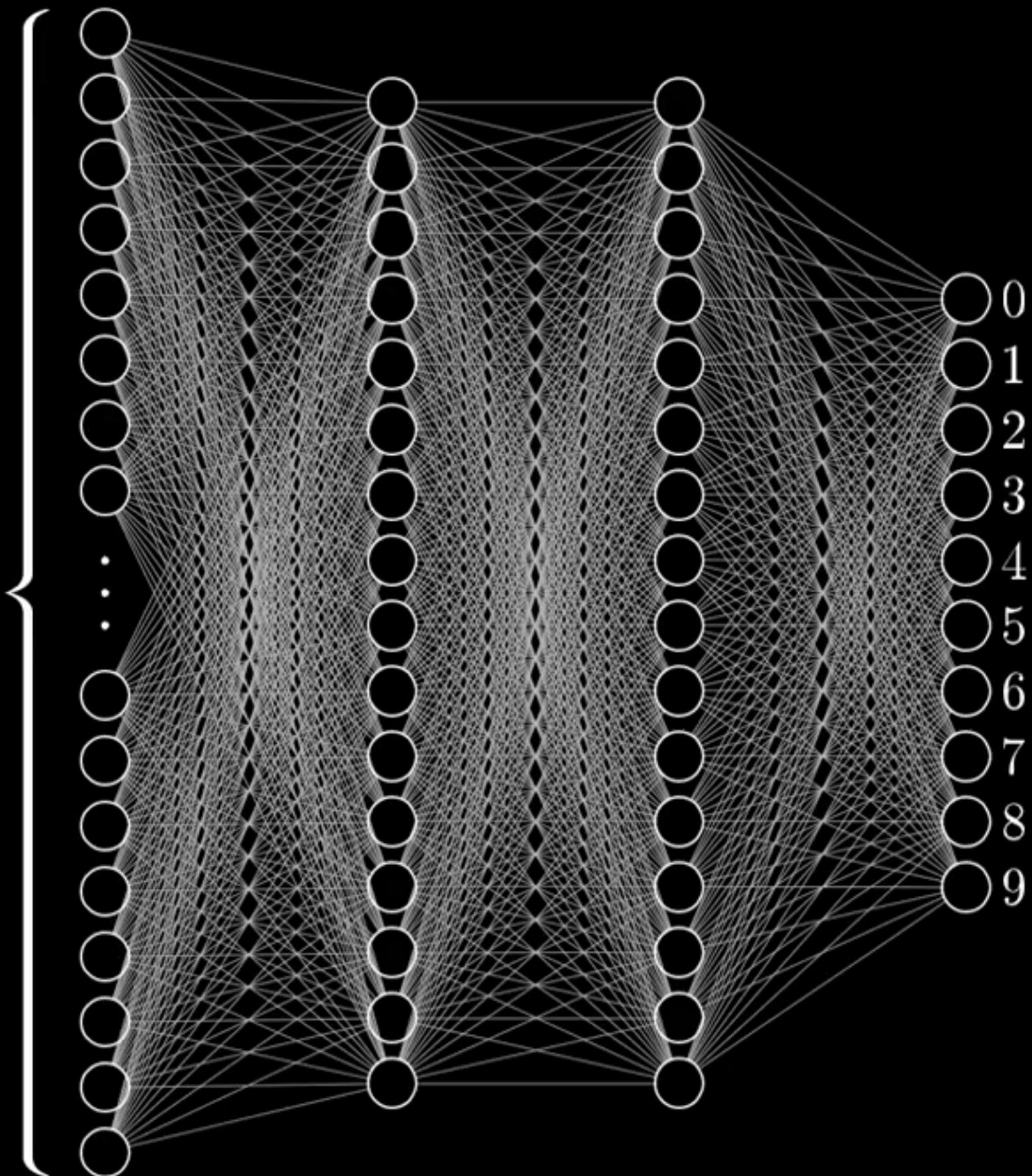


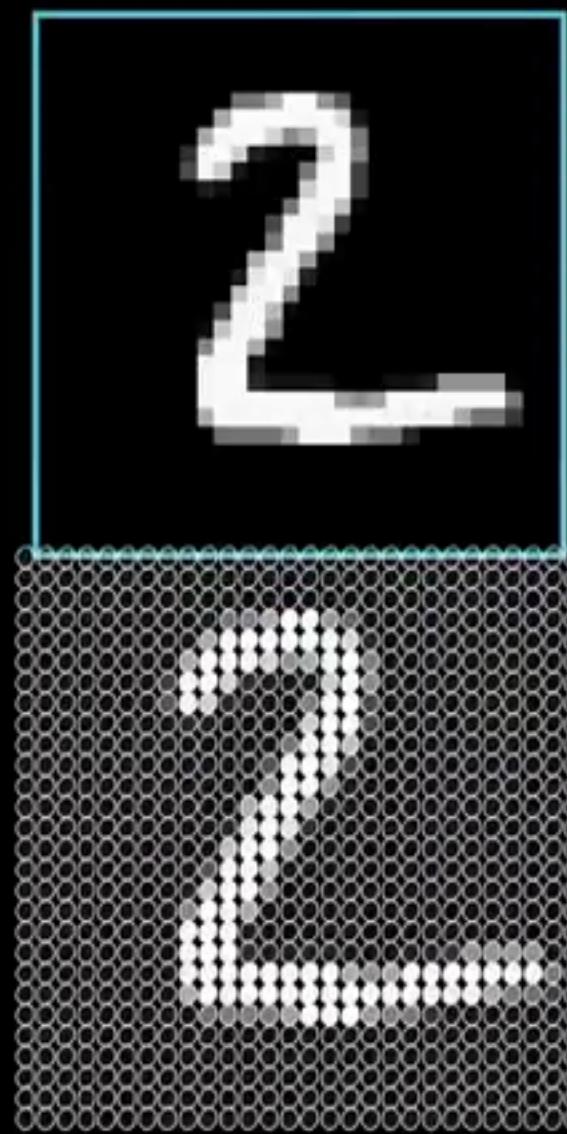
784



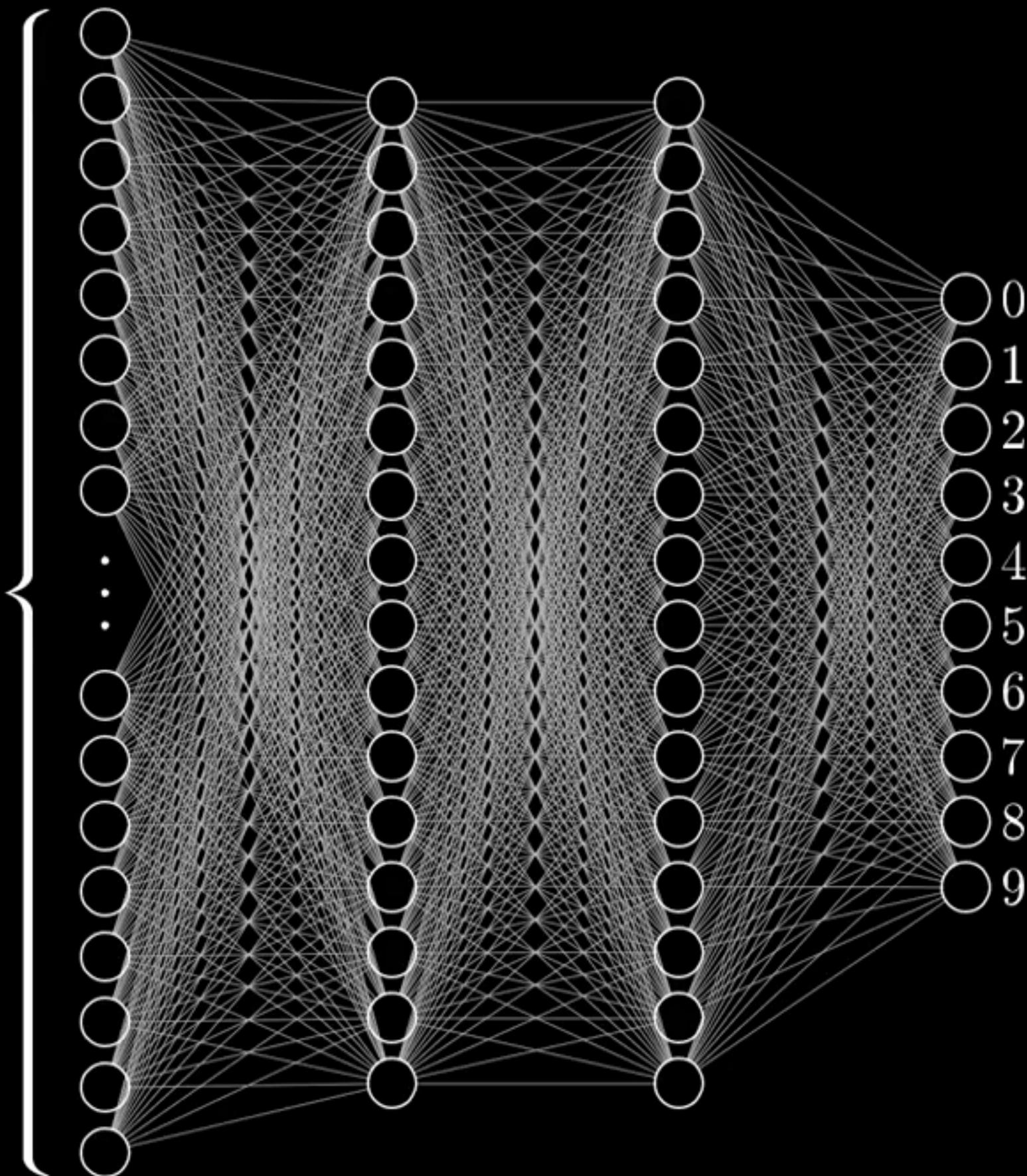


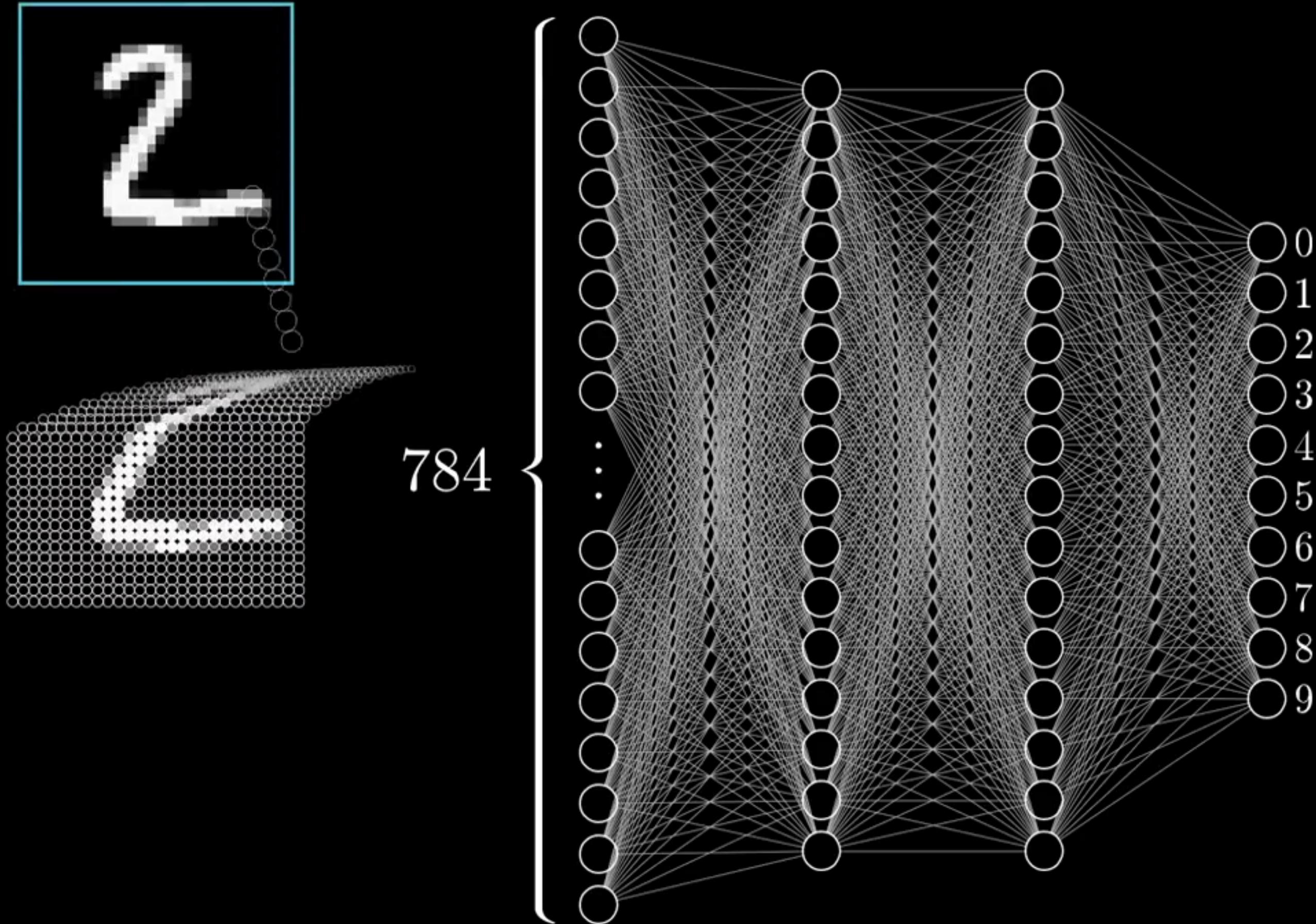
784





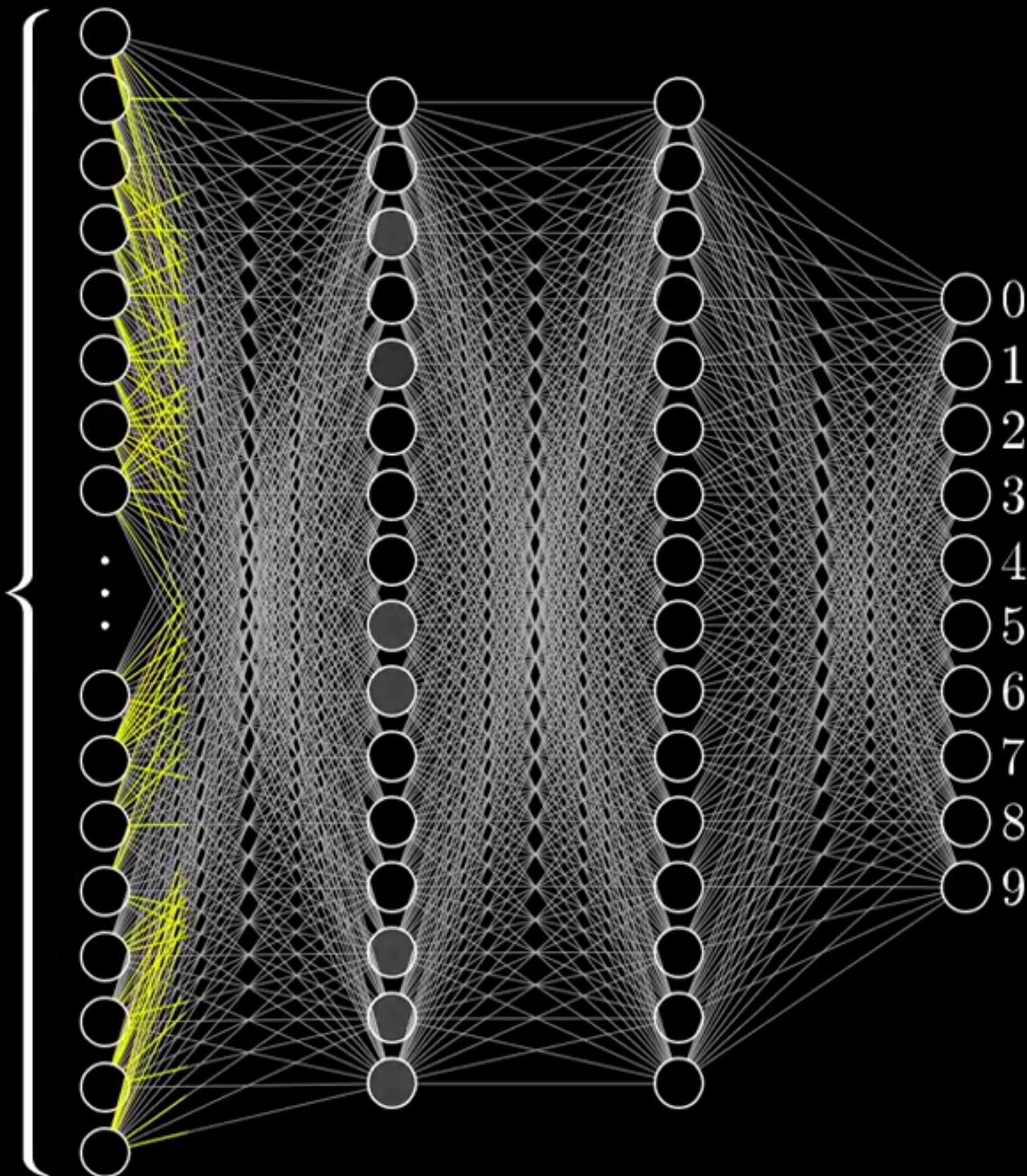
784





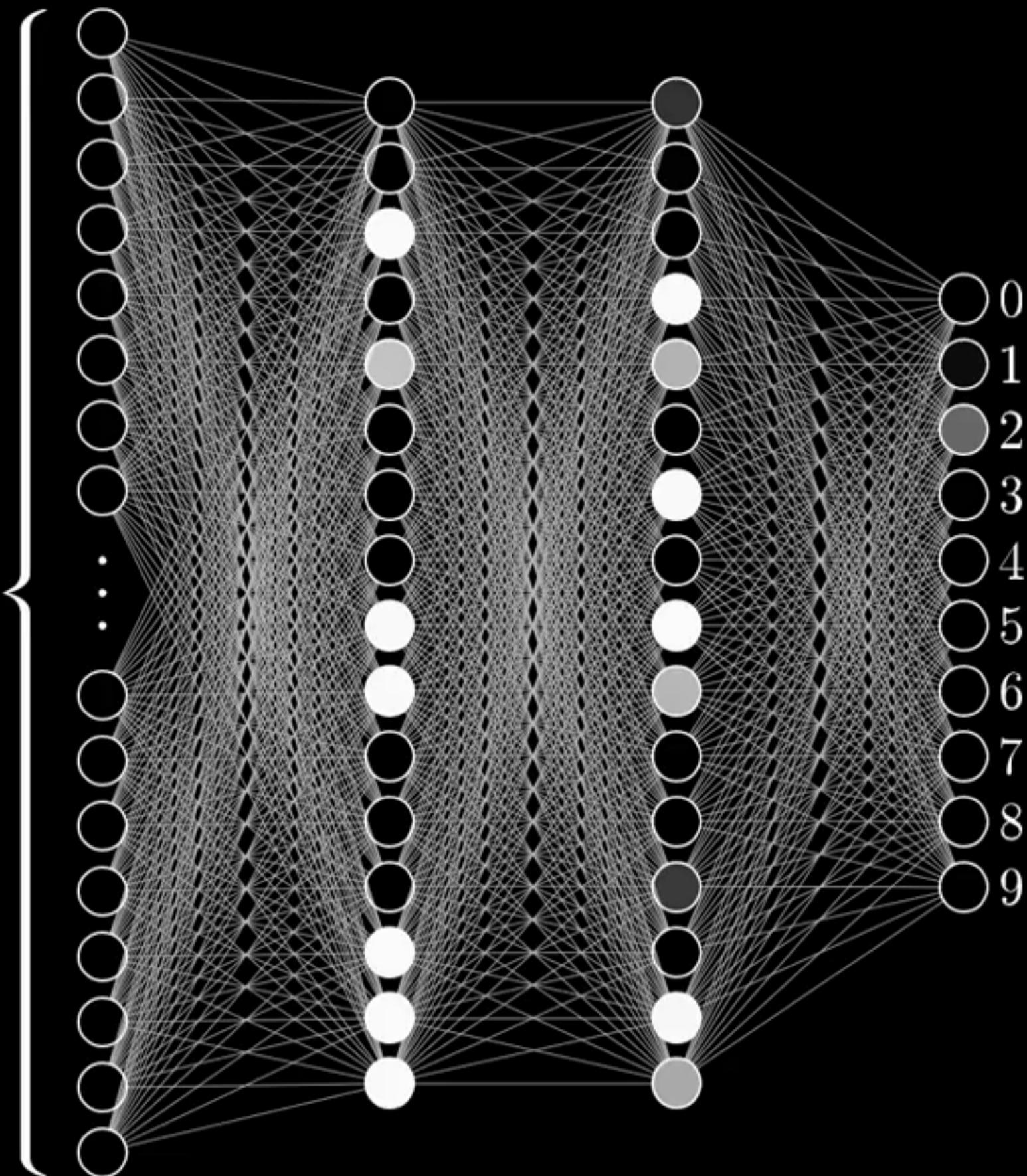


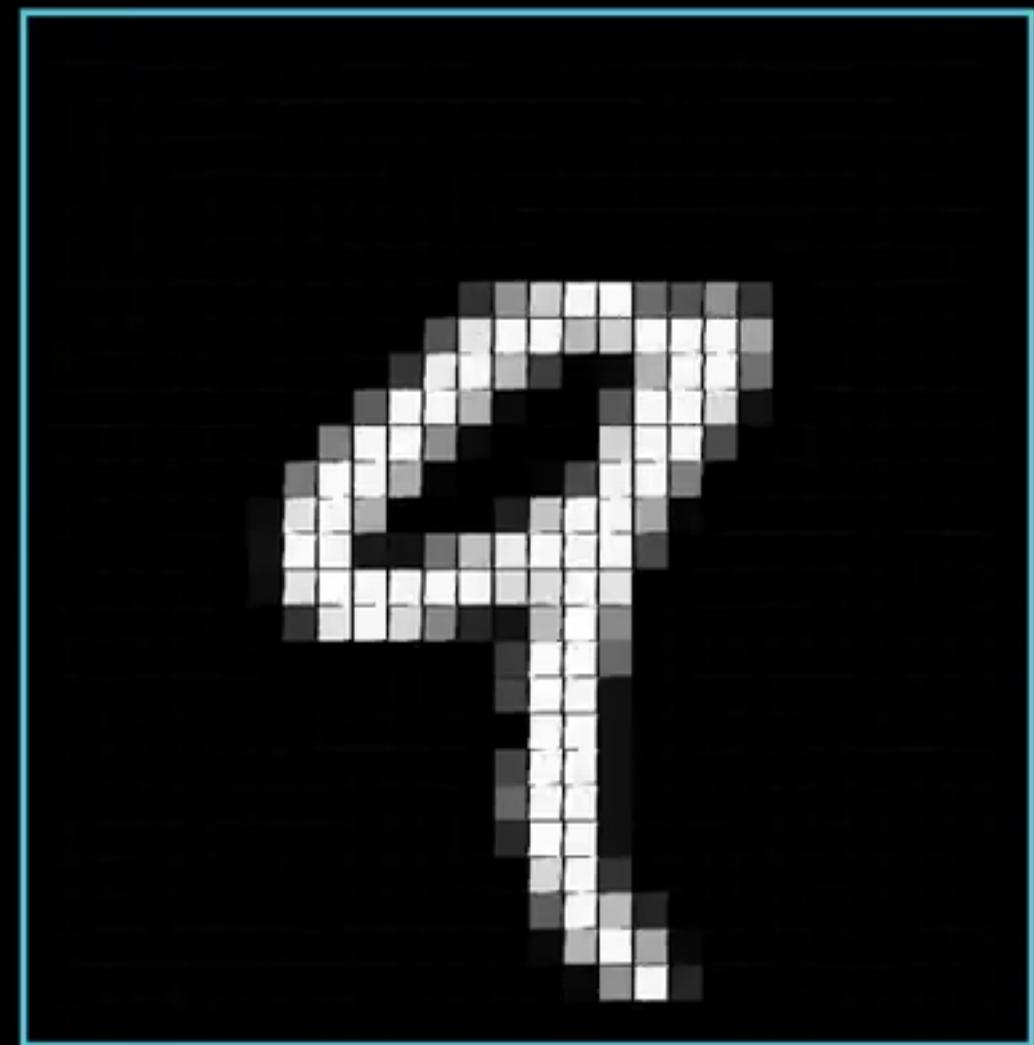
784





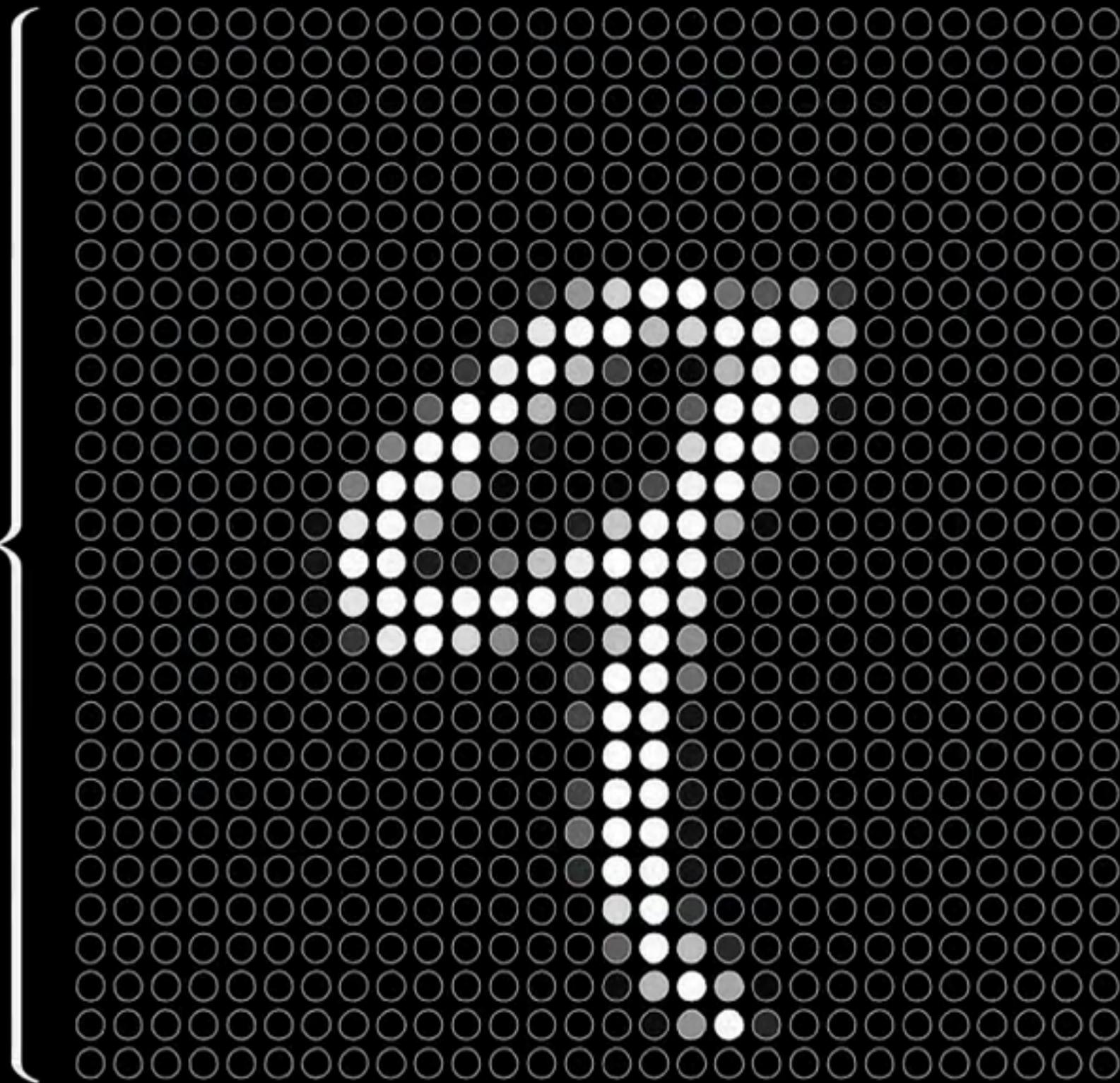
784





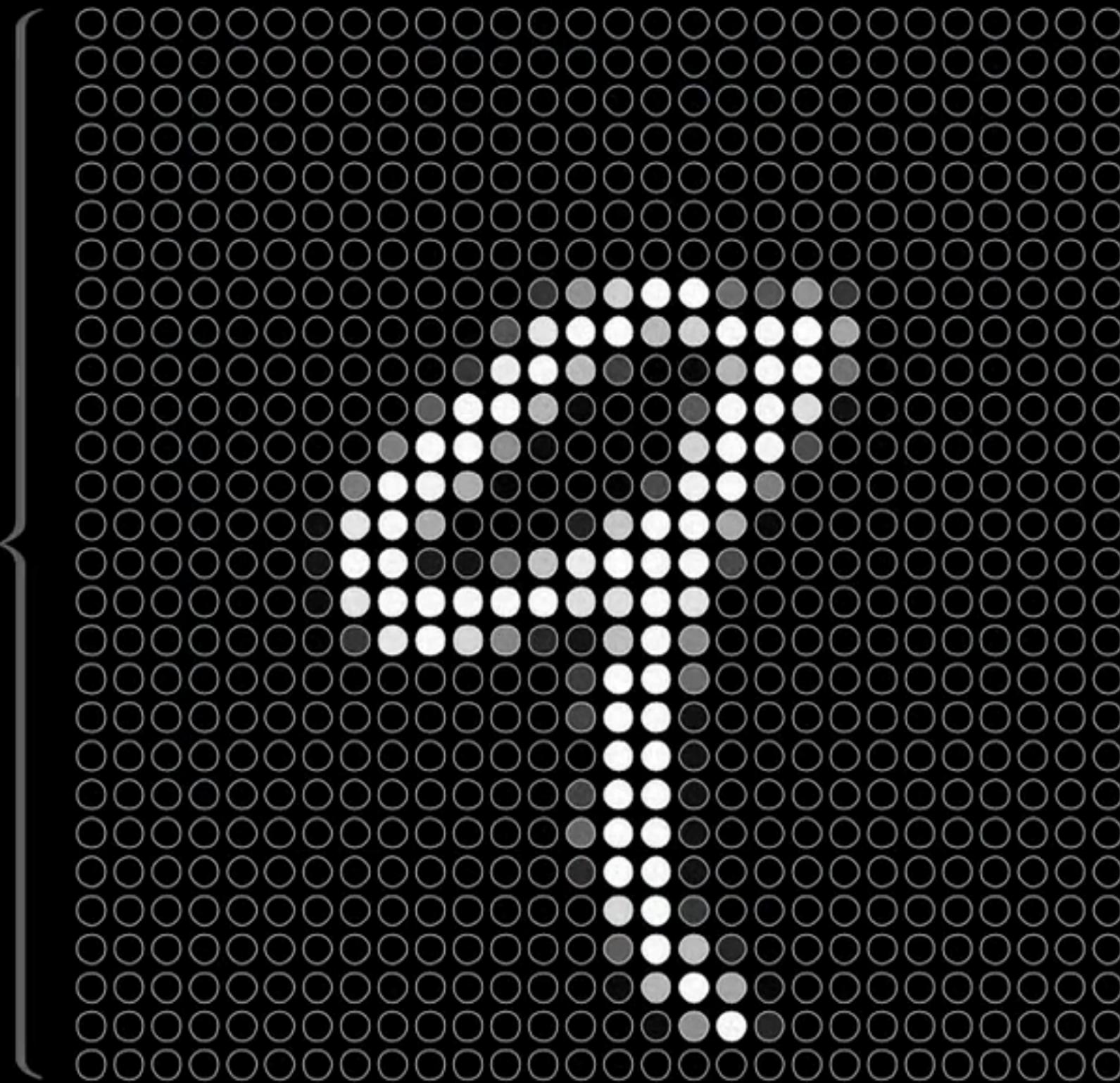
28

28



28

28

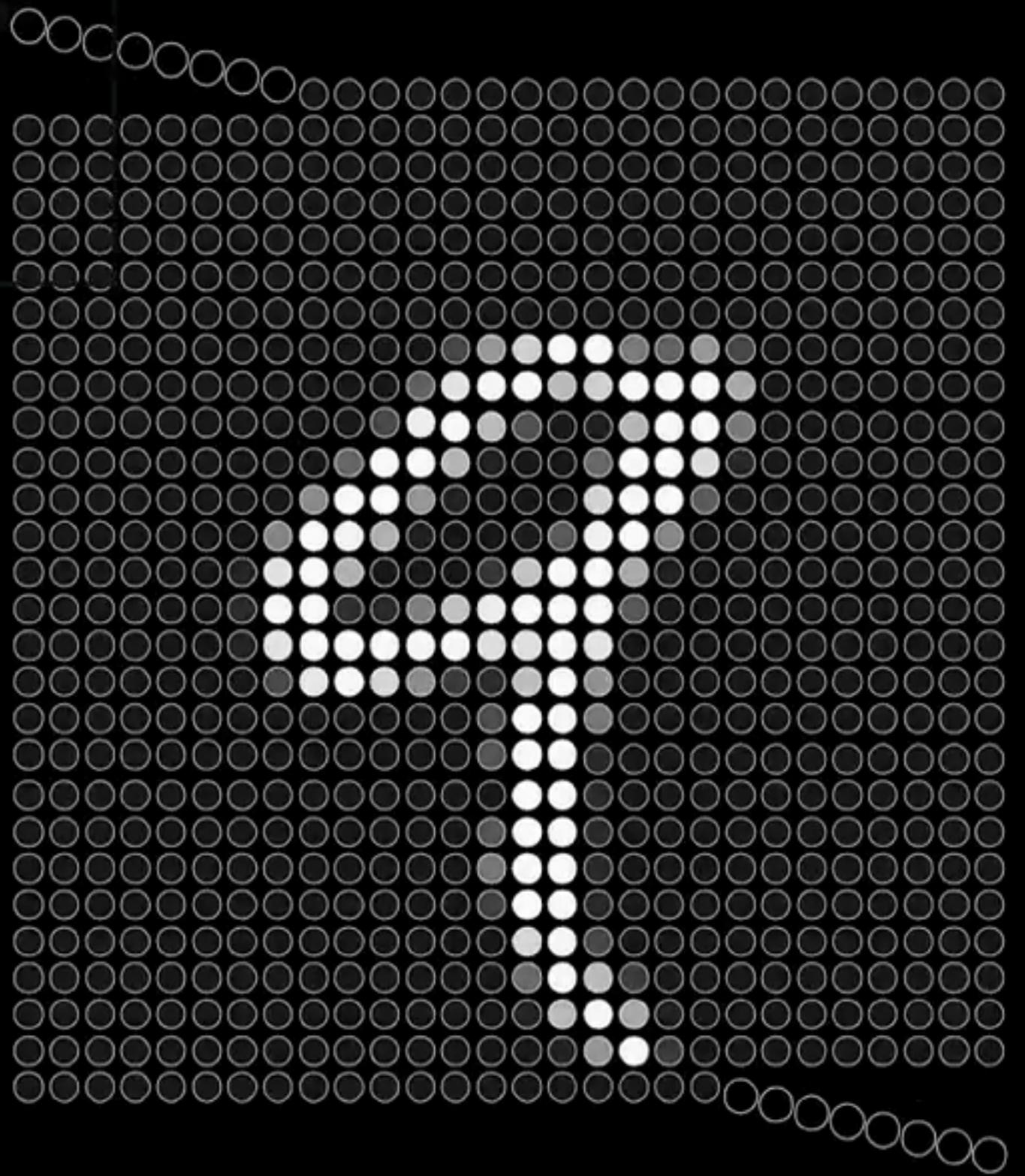


$$28 \times 28 = 784$$



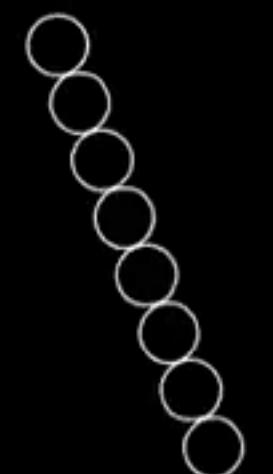
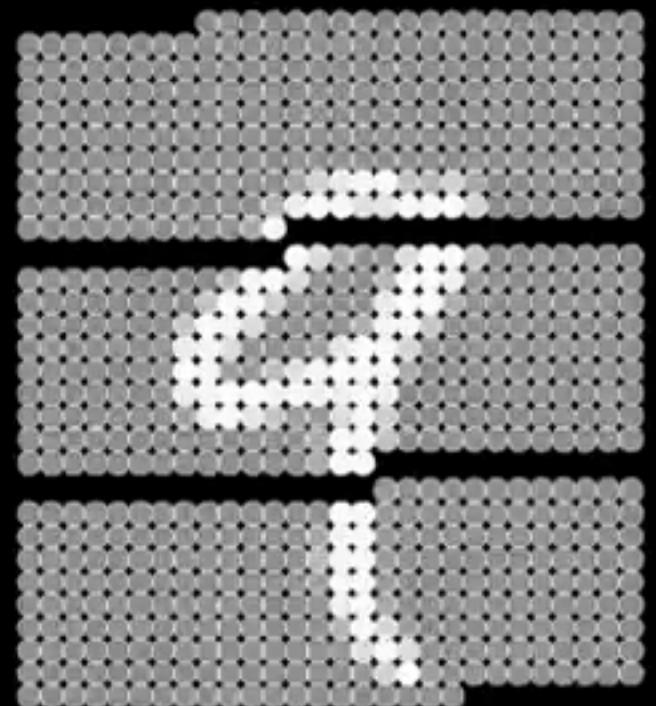
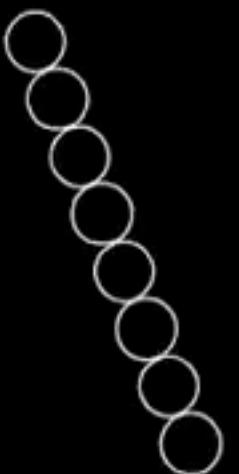
784

784



4

9



784

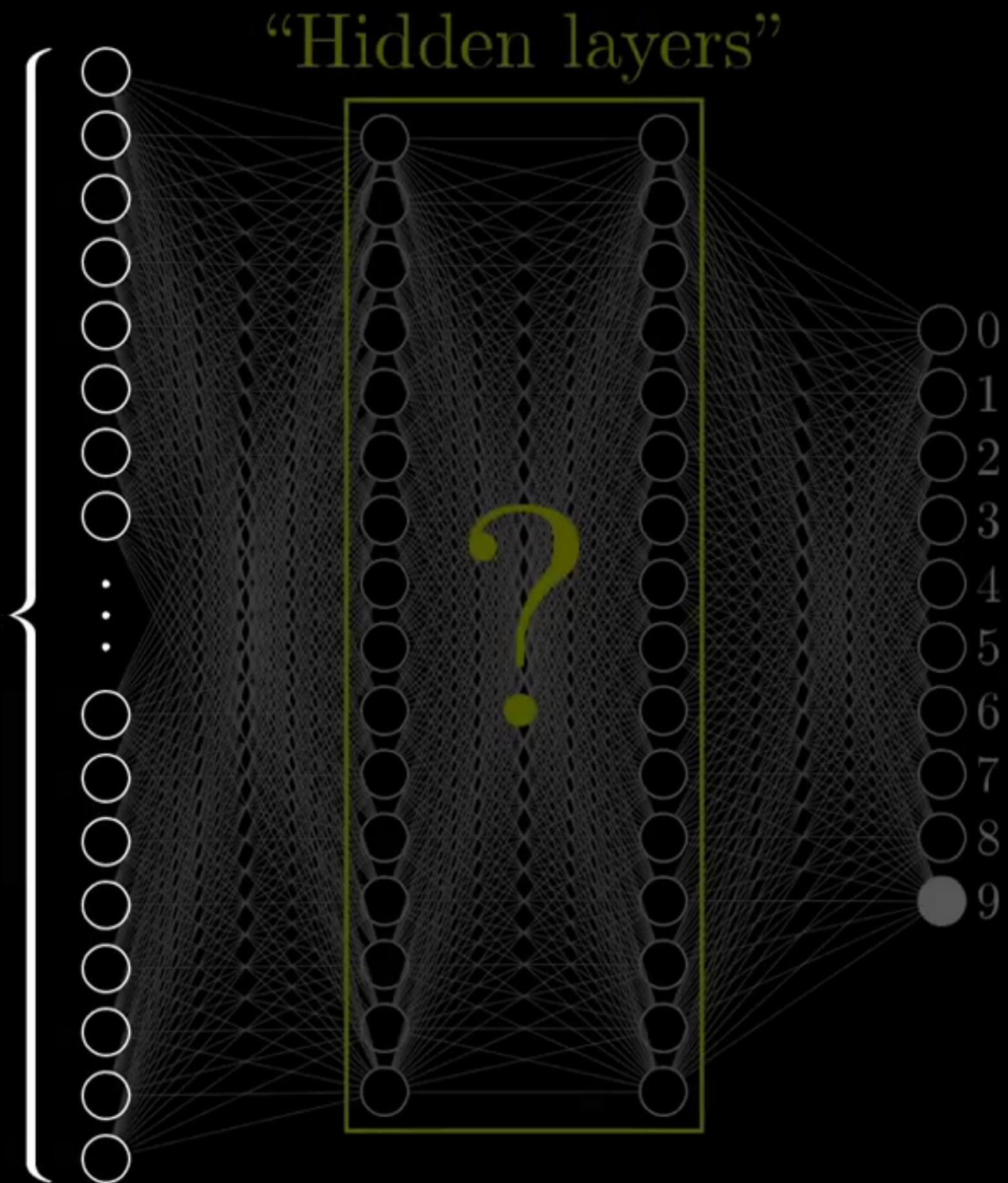


784



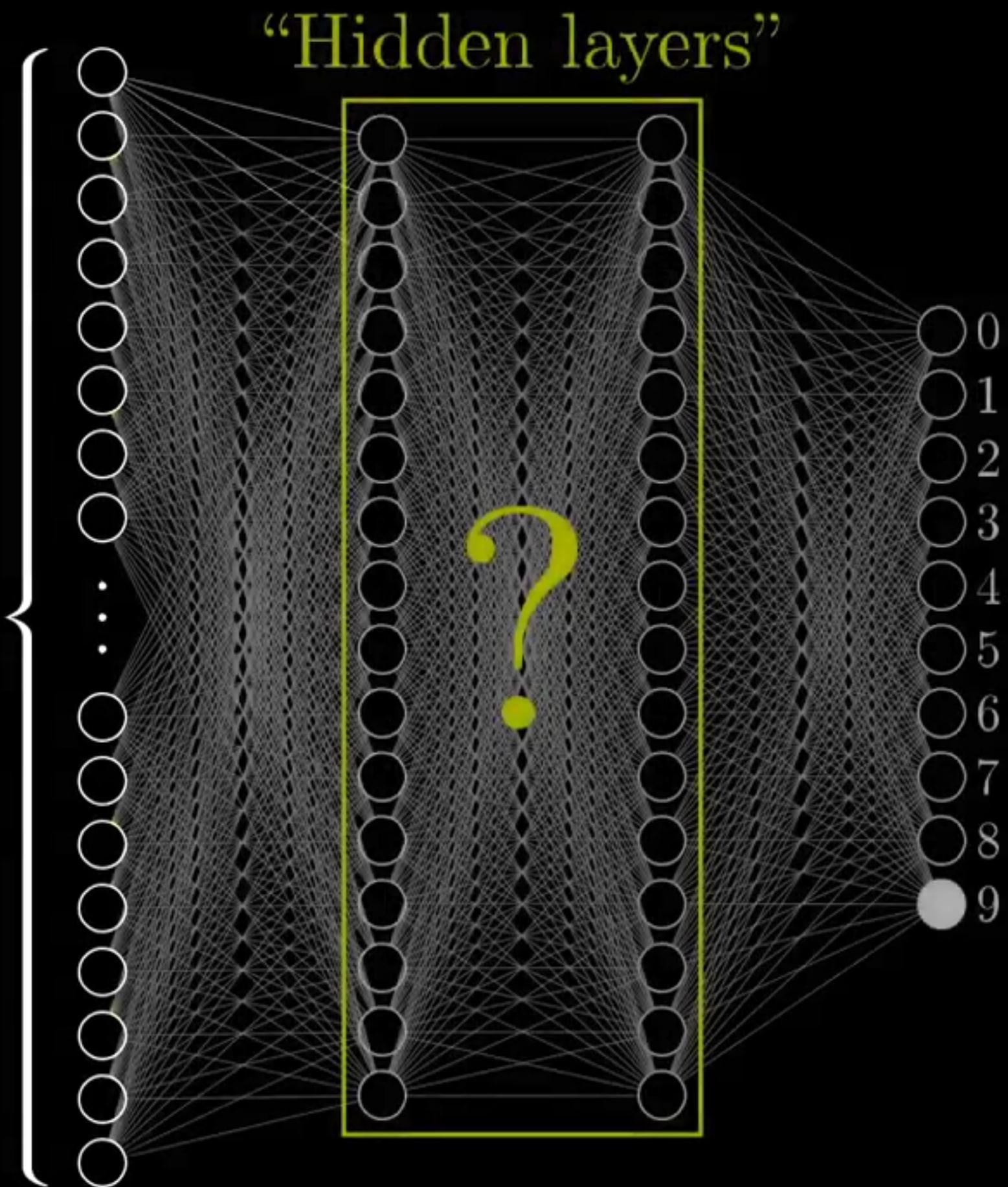


784



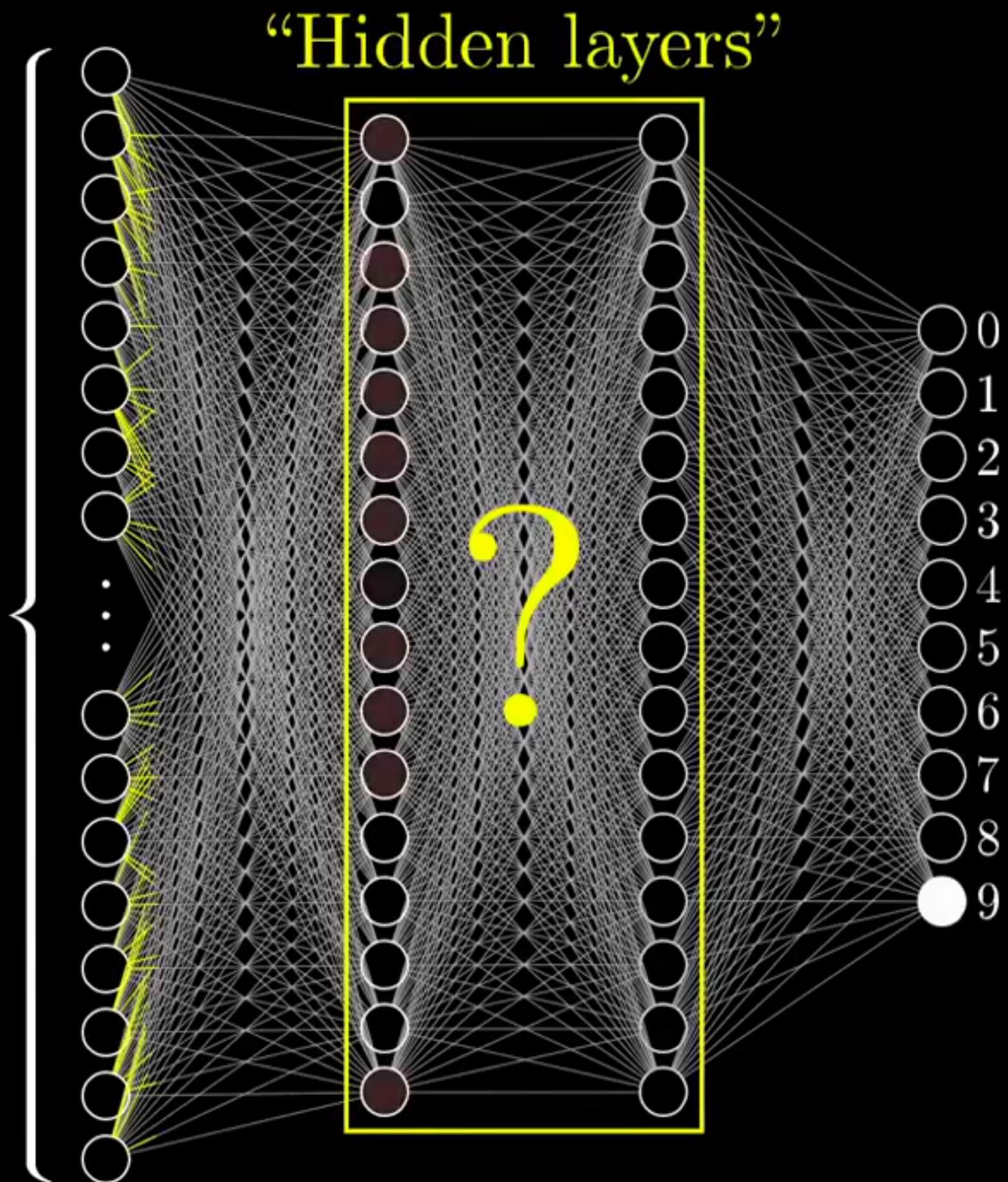


784



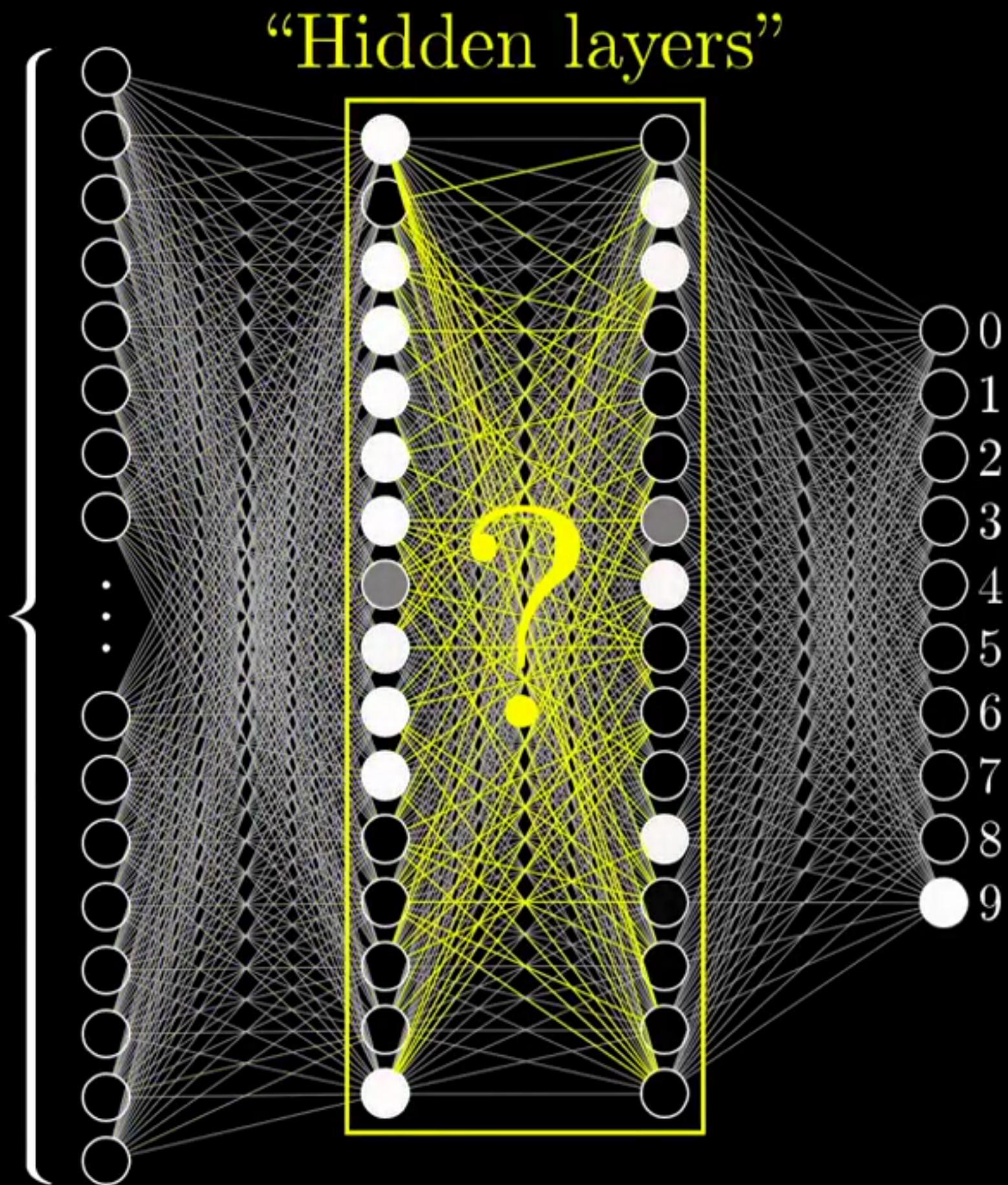


784



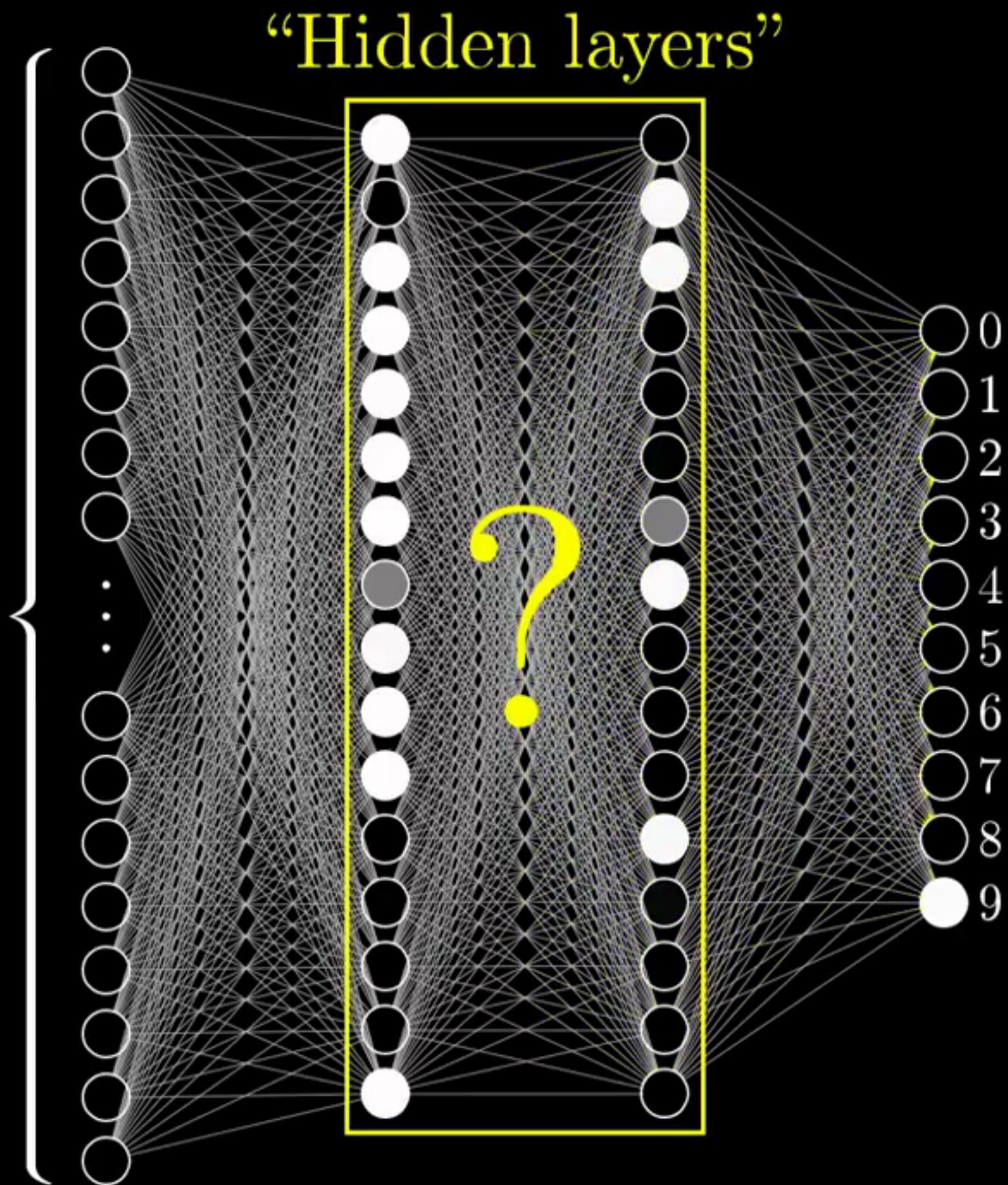


784



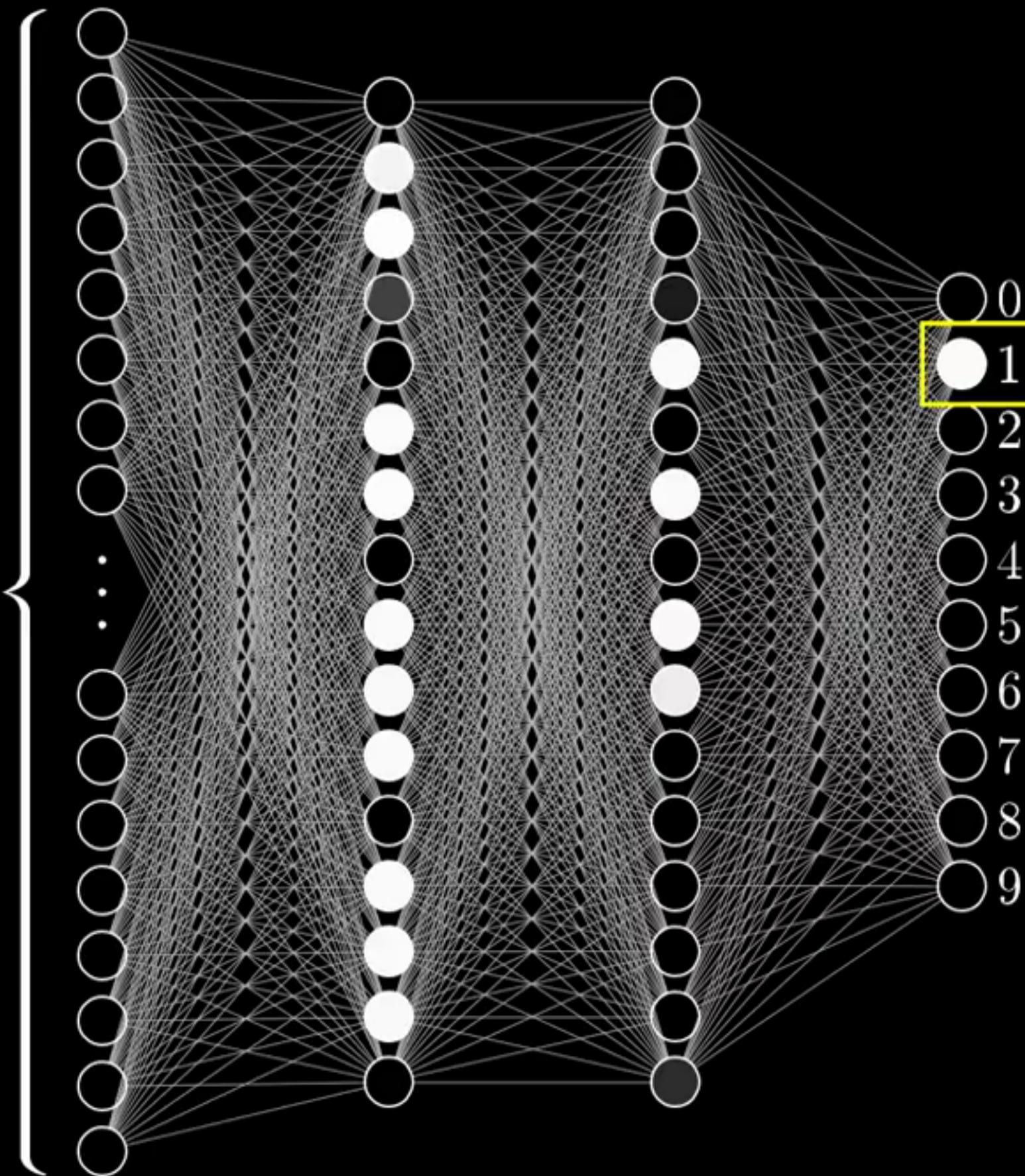


784



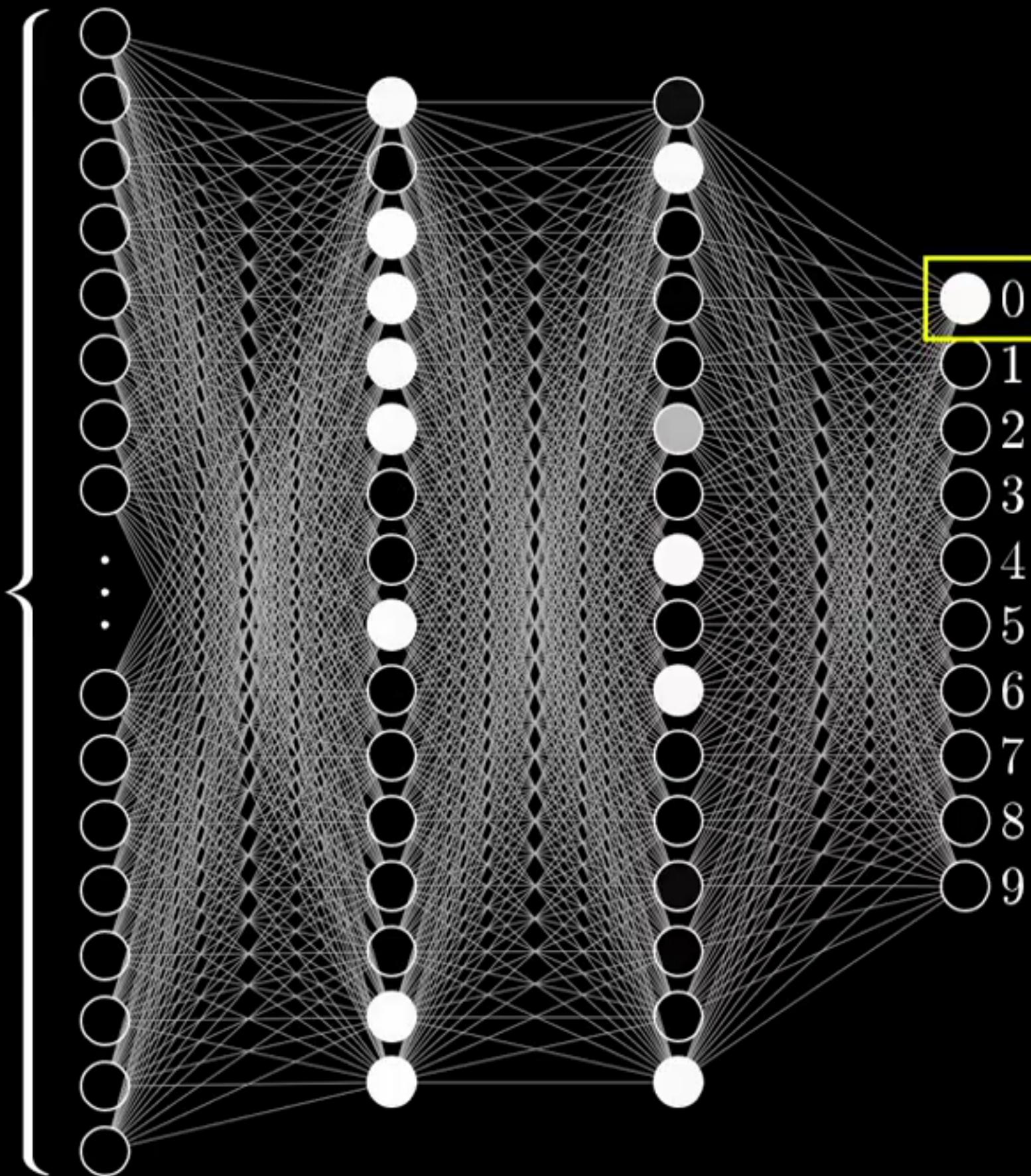


784



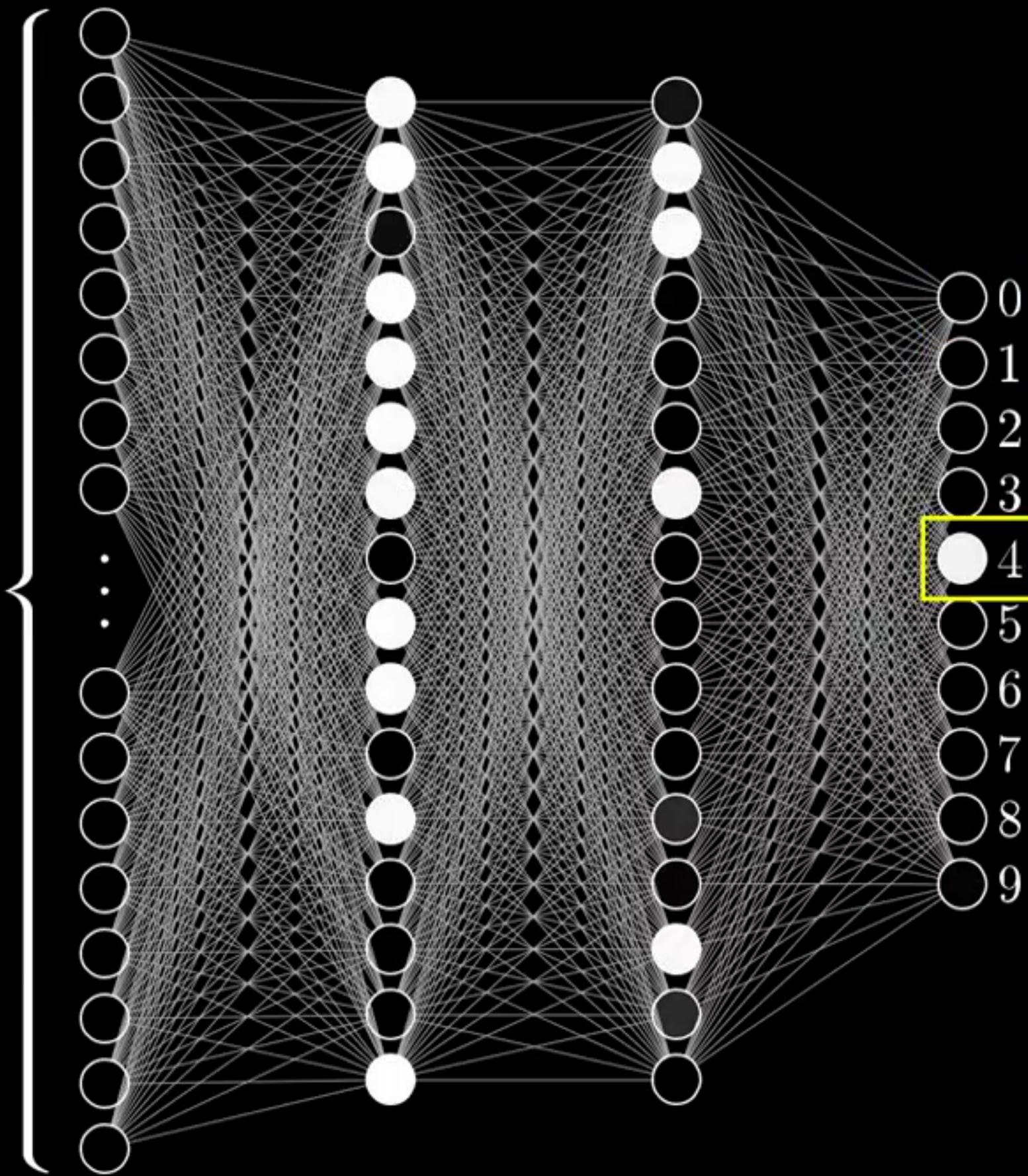


784



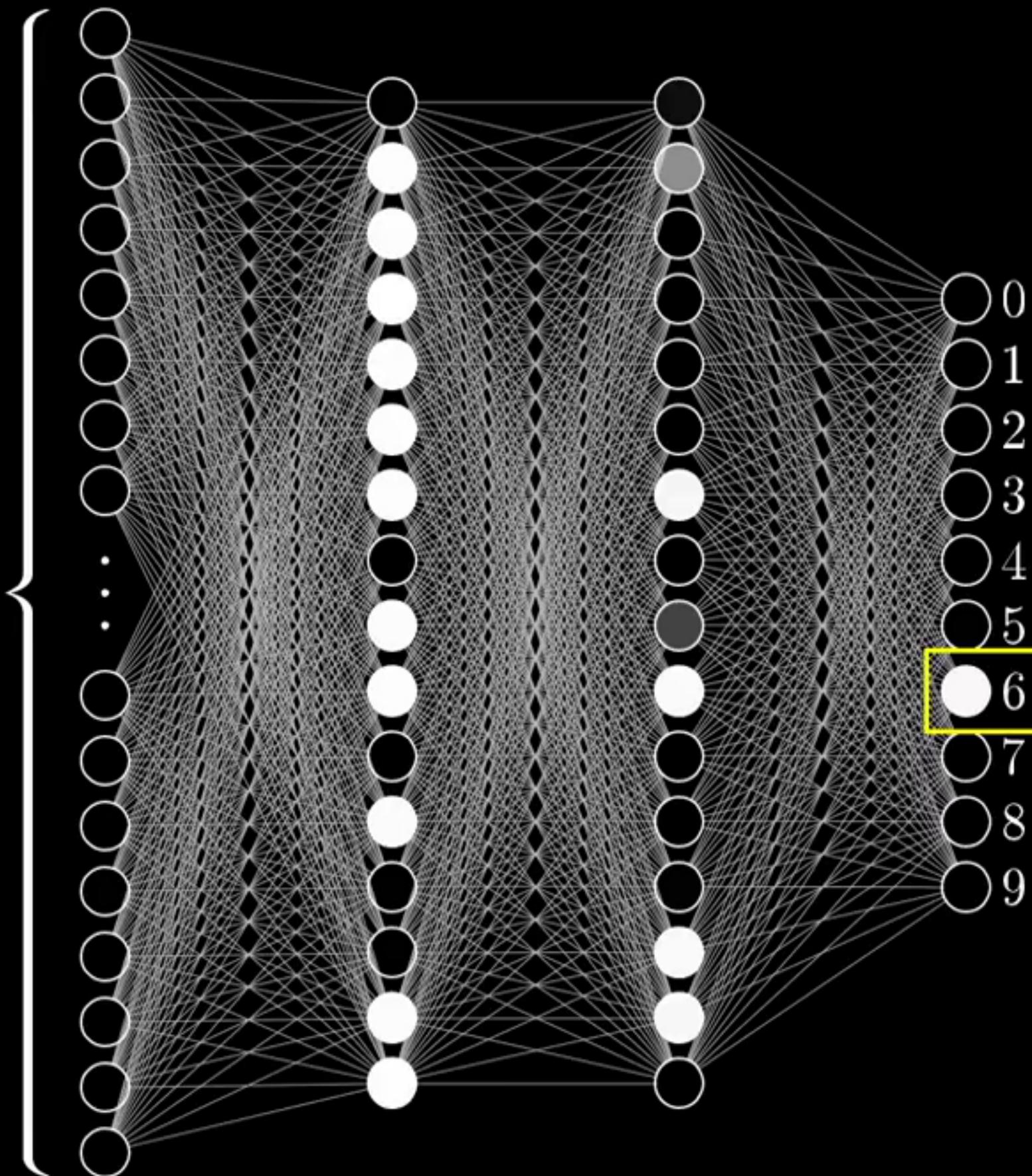


784



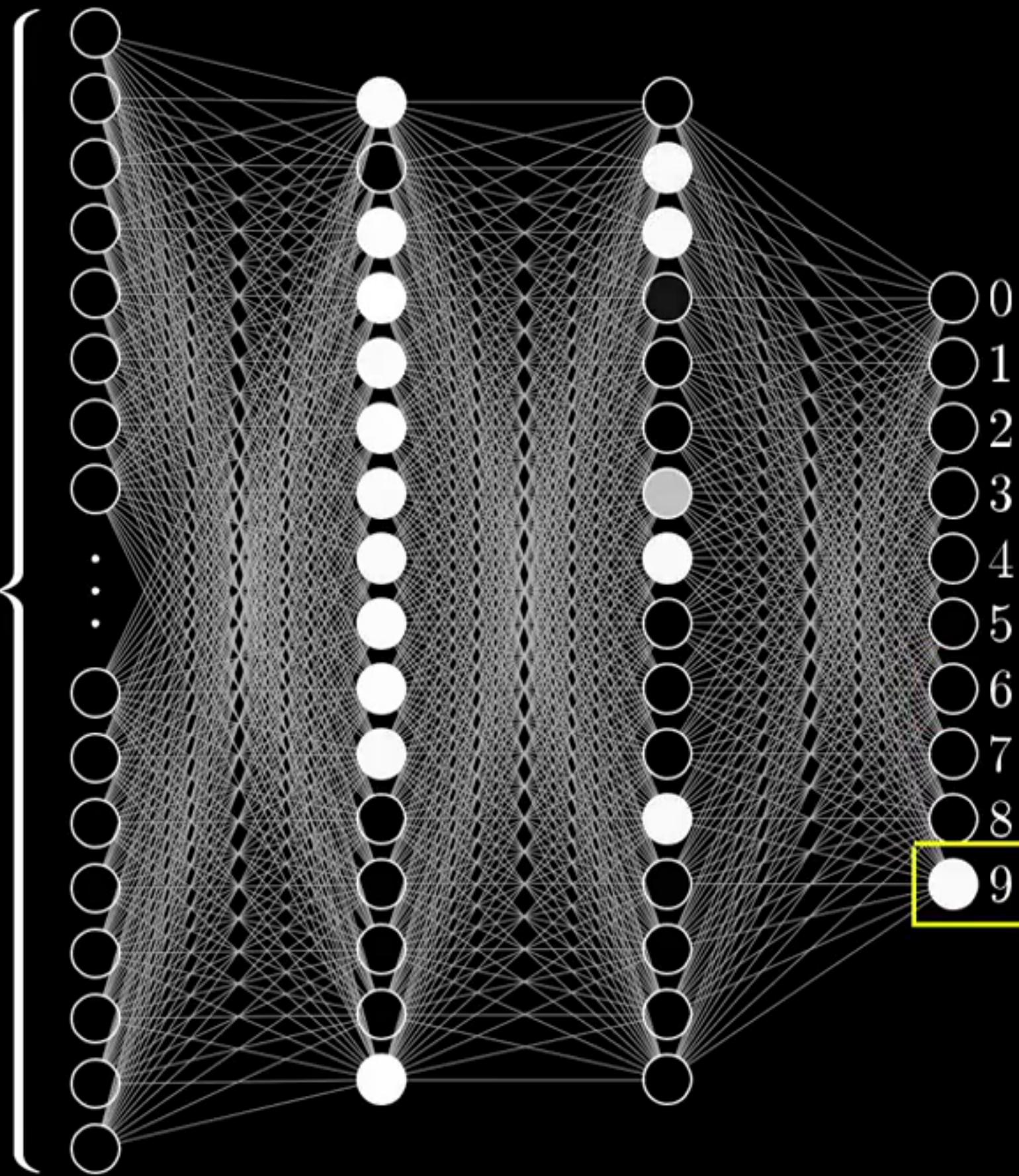


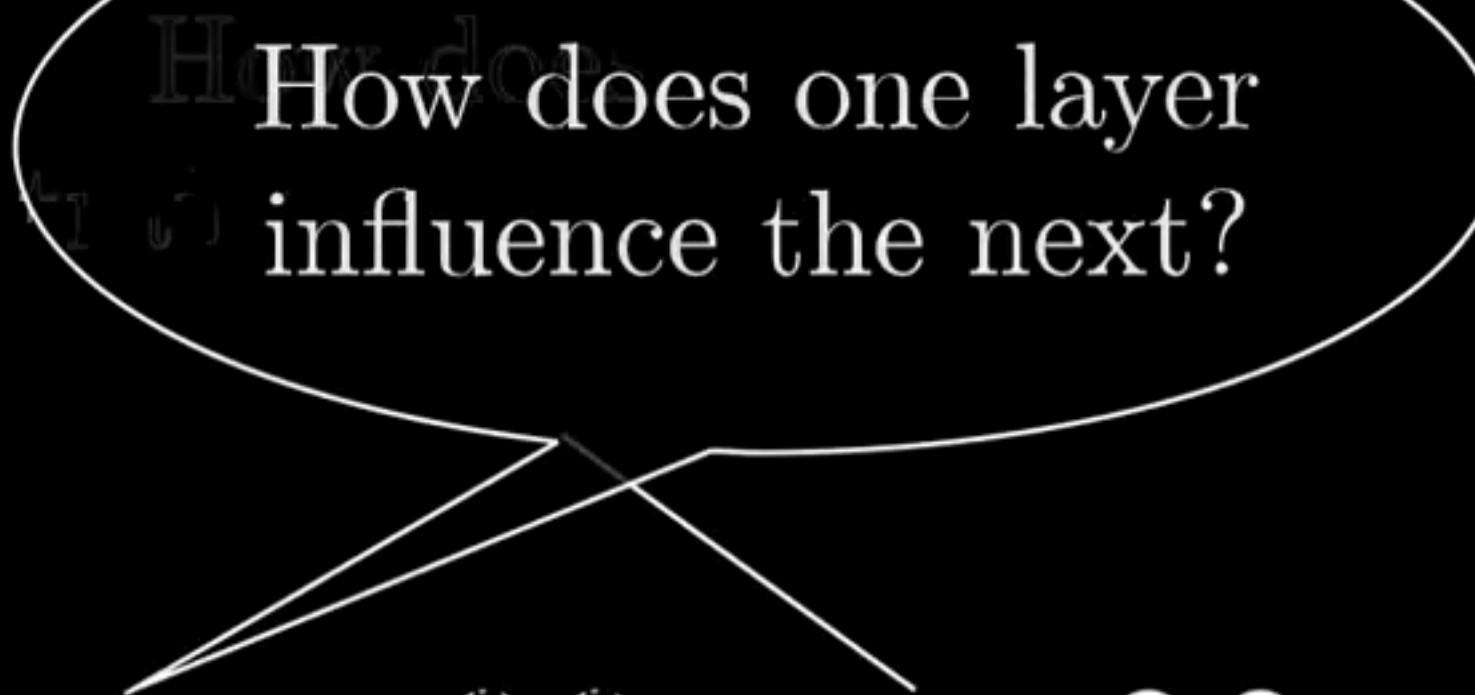
784



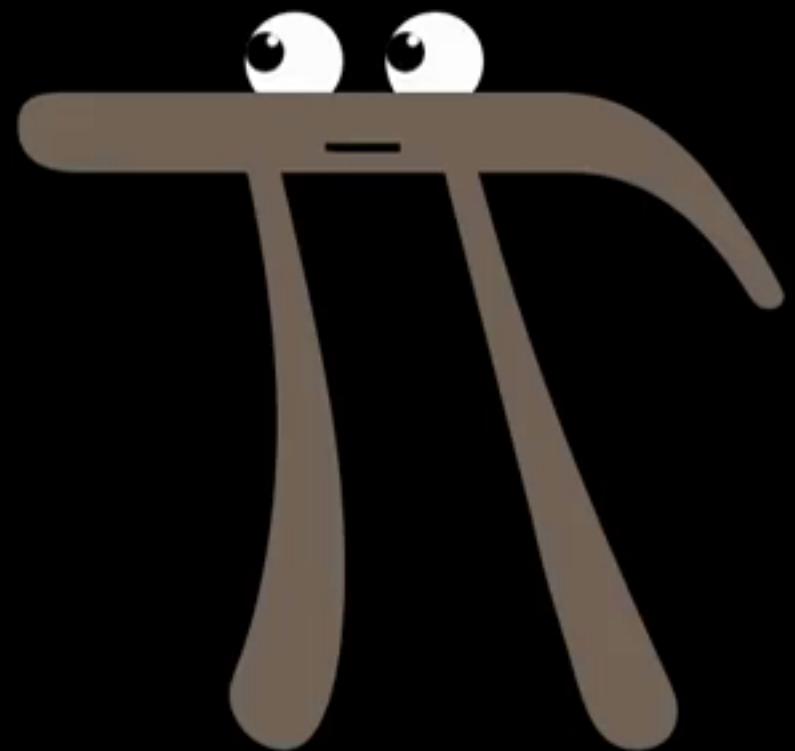
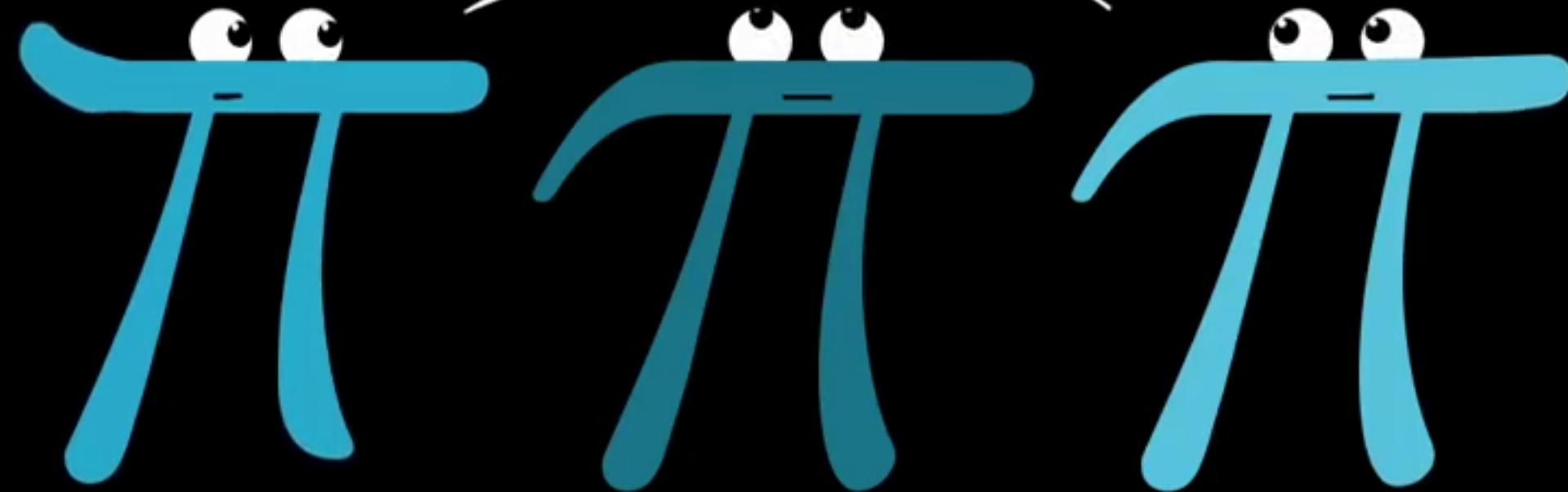


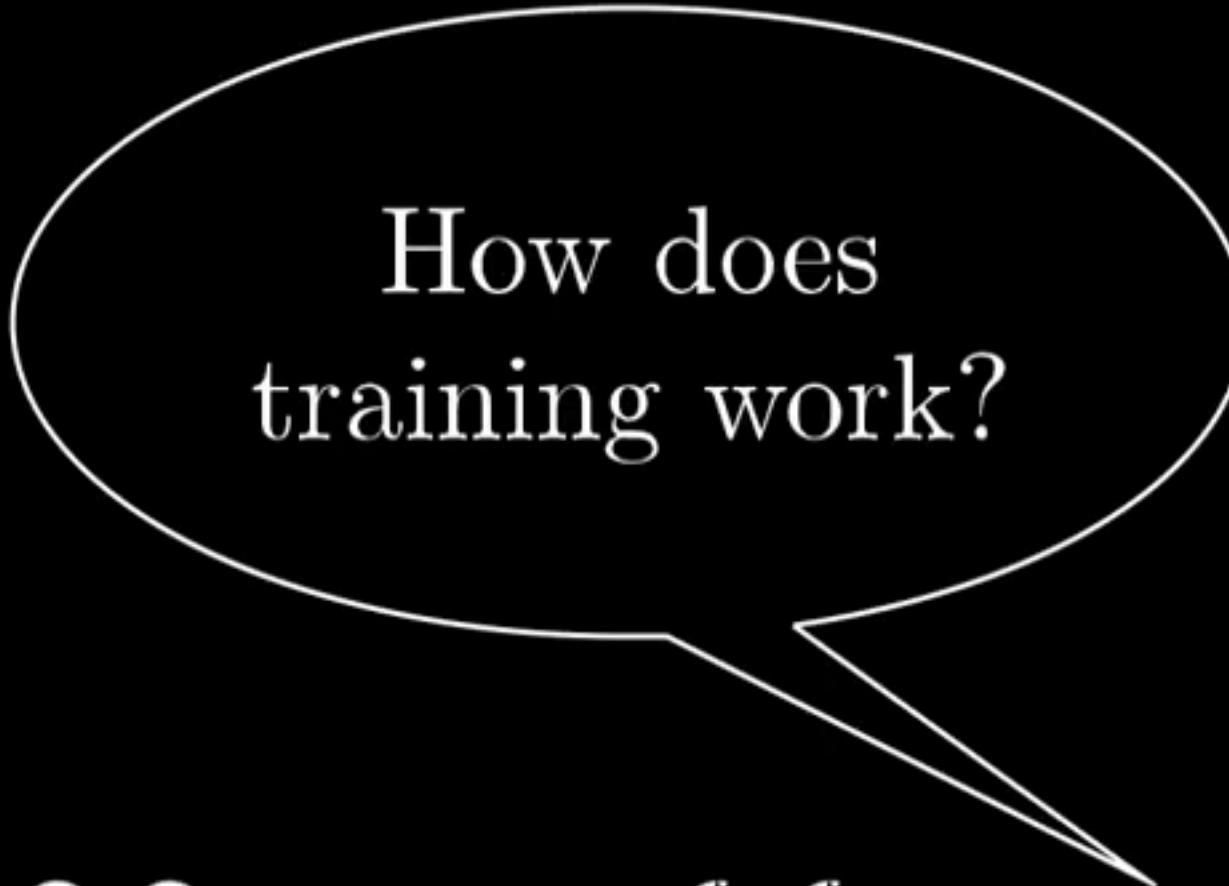
784



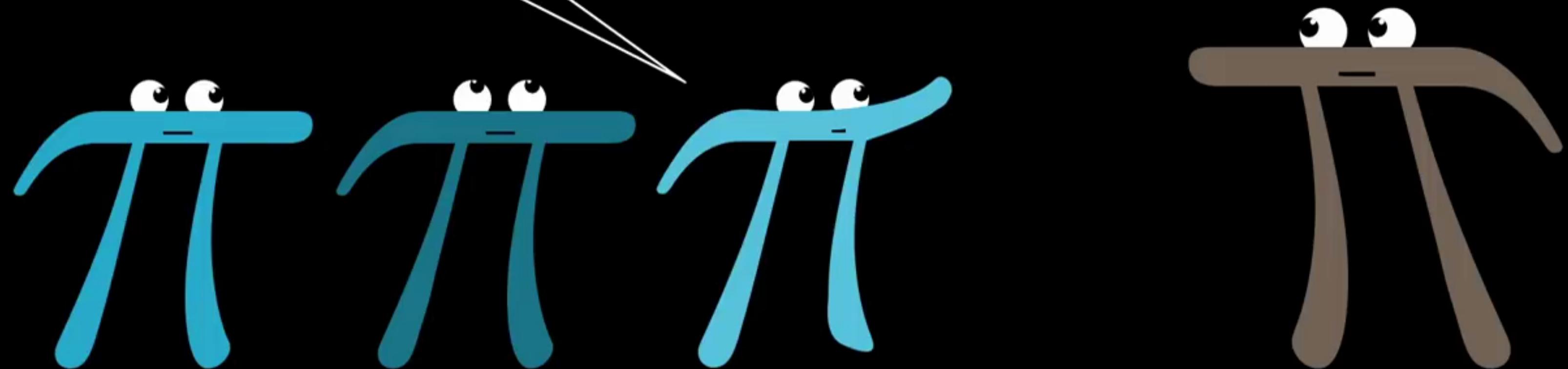


How does one layer
influence the next?

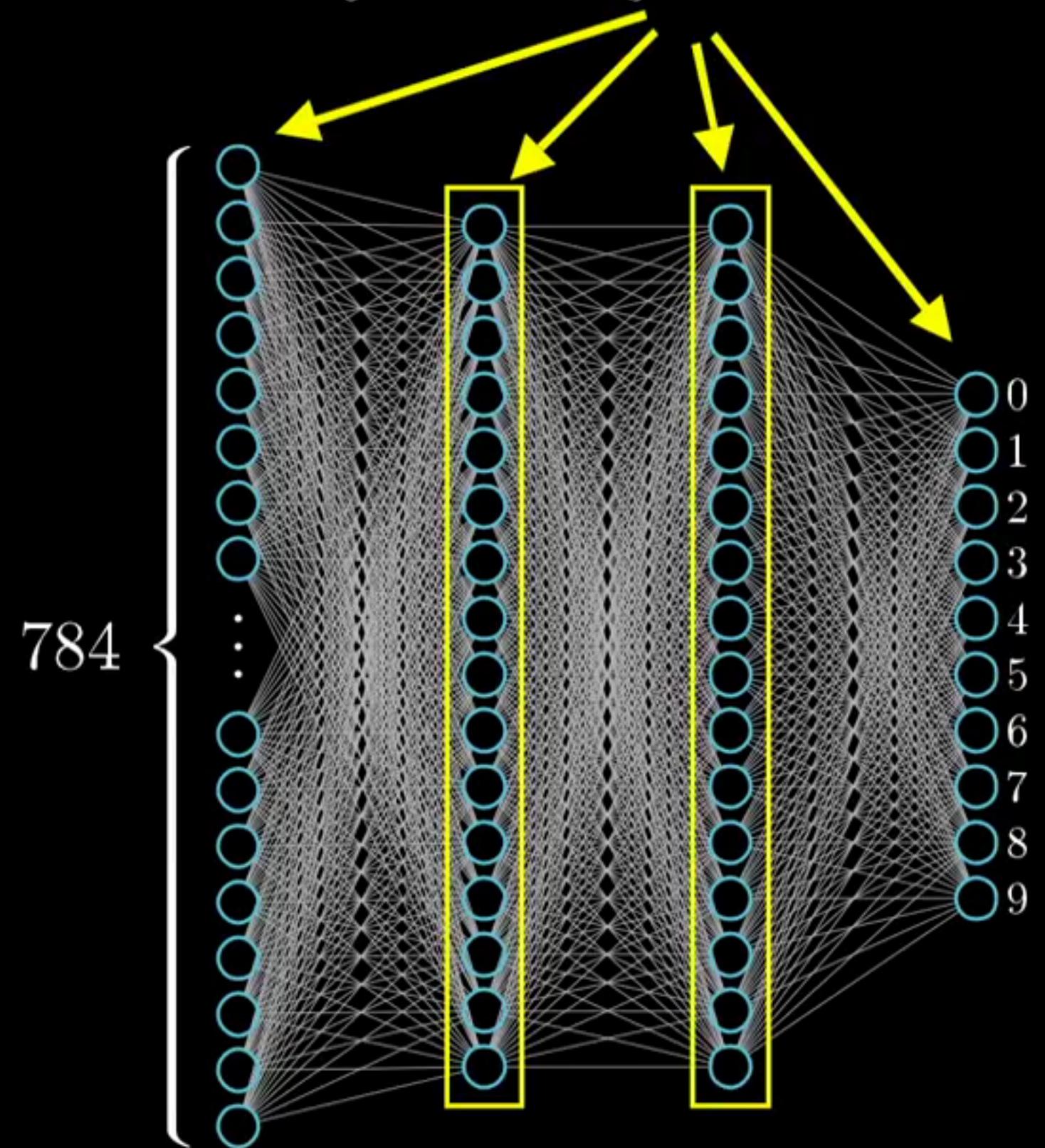




How does
training work?



Why the layers?



- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

$$\begin{matrix} \text{?} \\ \text{?} \end{matrix} = \begin{matrix} \text{?} \\ \text{?} \end{matrix} + \begin{matrix} \text{?} \end{matrix}$$

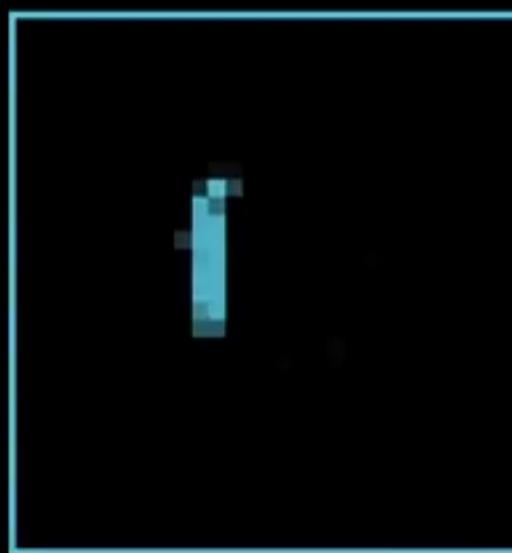
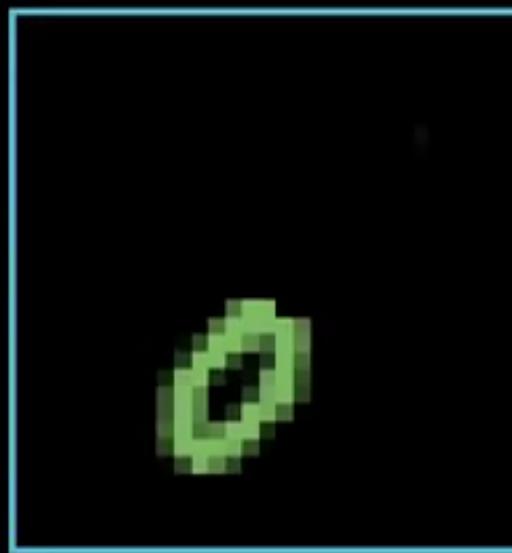
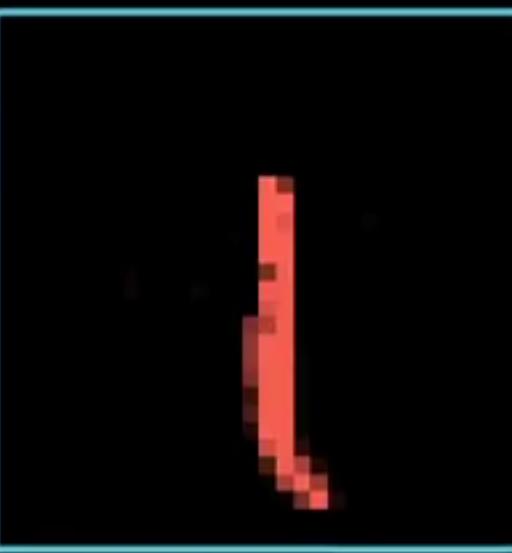
$$\begin{matrix} \text{?} \\ \text{?} \end{matrix} = \begin{matrix} \text{?} \\ \text{?} \end{matrix} + \begin{matrix} \text{?} \end{matrix}$$

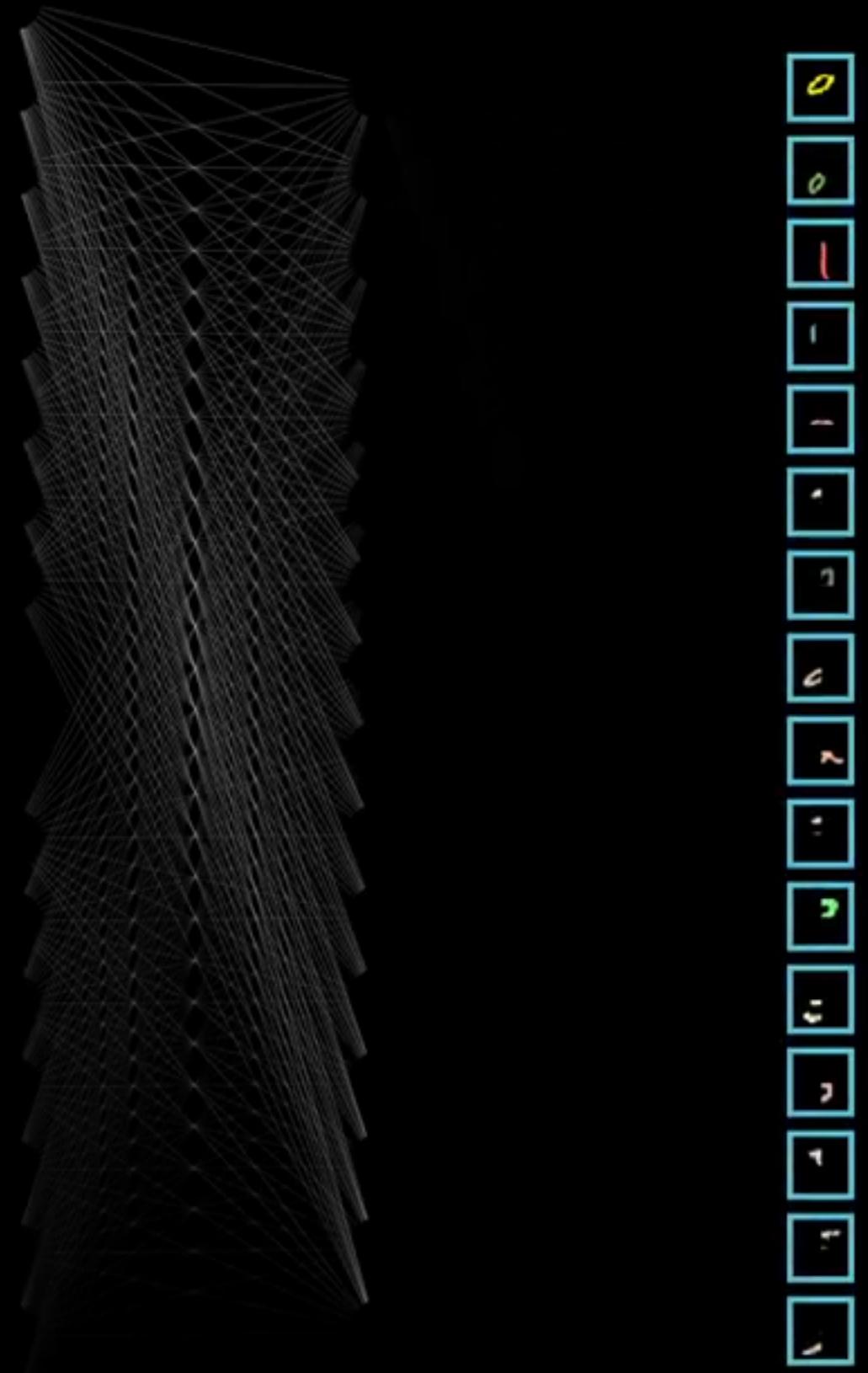
$$\begin{matrix} \text{?} \\ \text{?} \end{matrix} = \begin{matrix} \text{?} \\ \text{?} \end{matrix} + \begin{matrix} \text{?} \end{matrix} + \begin{matrix} \text{?} \end{matrix}$$

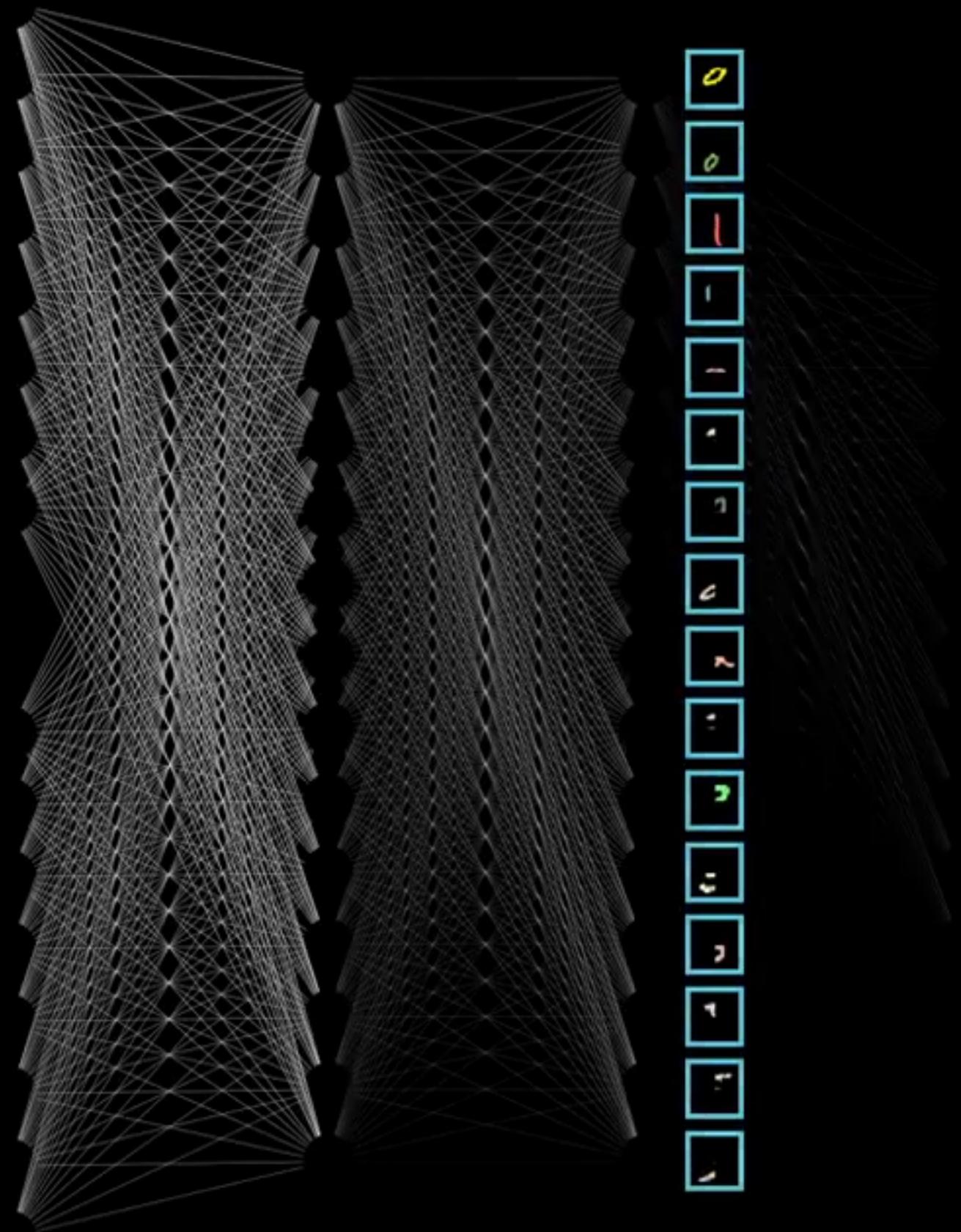
$$\varphi = \text{yellow blob} + \text{red vertical bar}$$

$$\varphi' = \text{green blob} + \text{green blob}$$

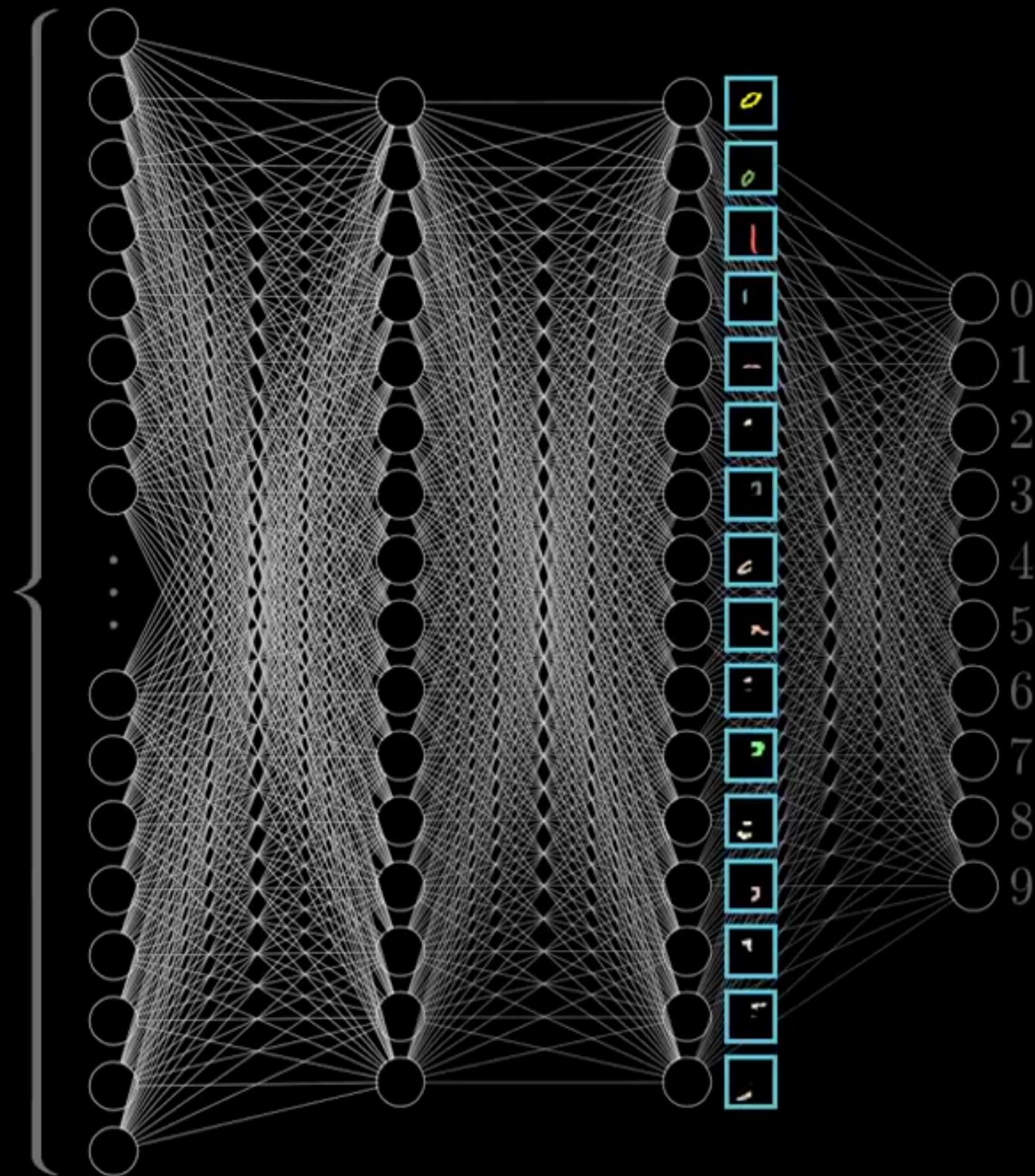
$$\varphi = \text{dark red vertical bar} + \text{cyan vertical bar} + \text{pink horizontal bar}$$







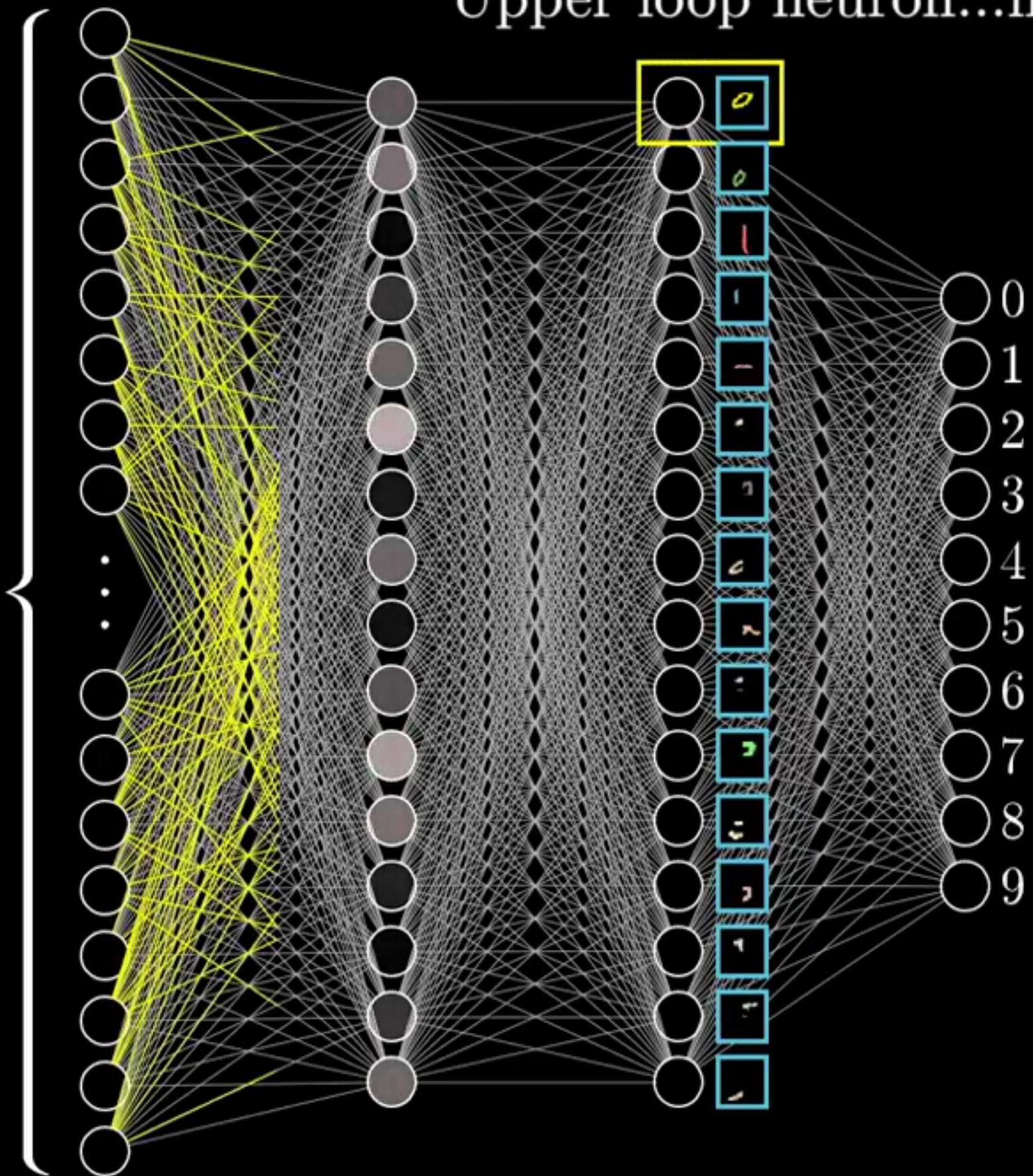
784



Upper loop neuron...maybe...



784

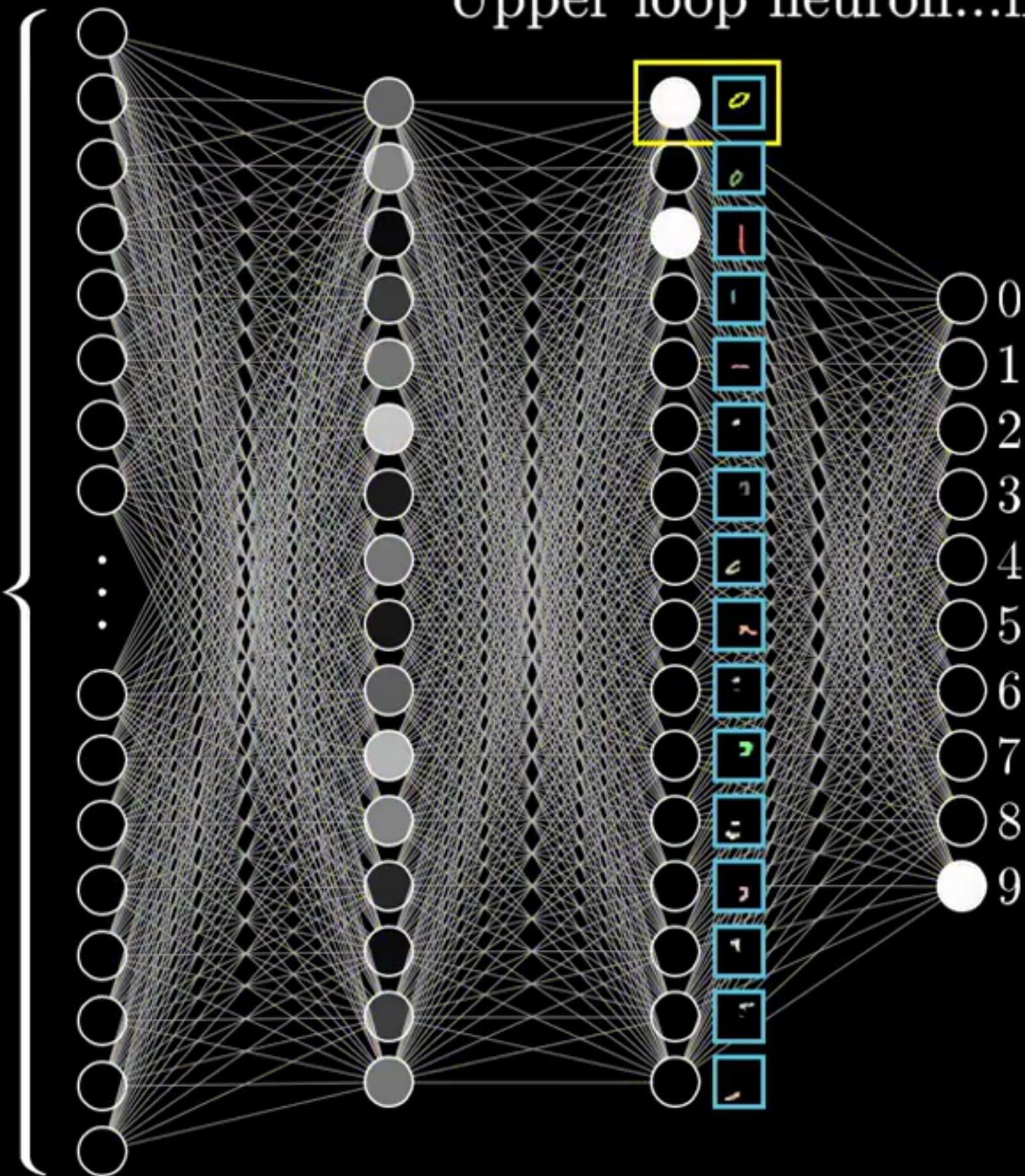


- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

Upper loop neuron...maybe...



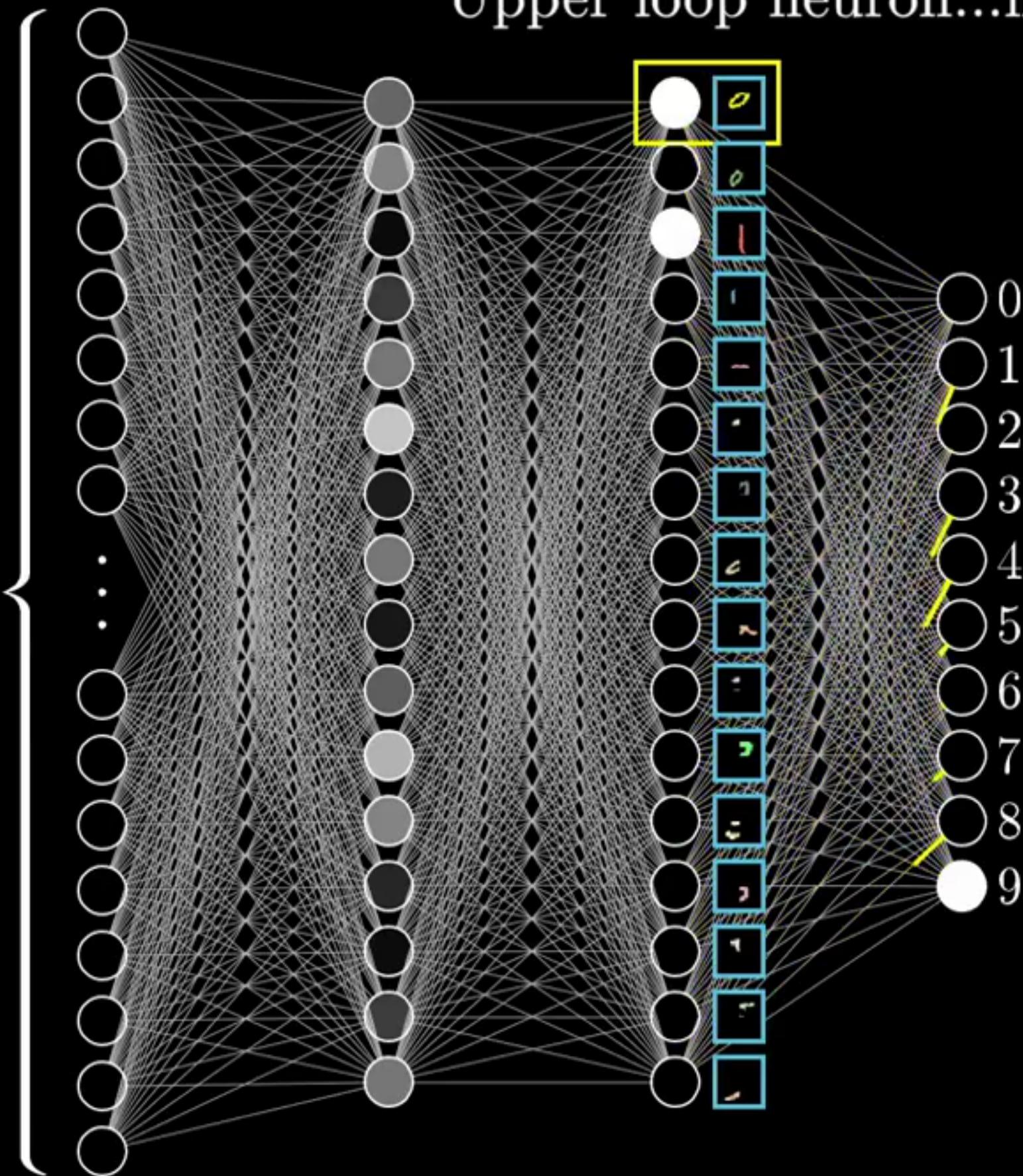
784

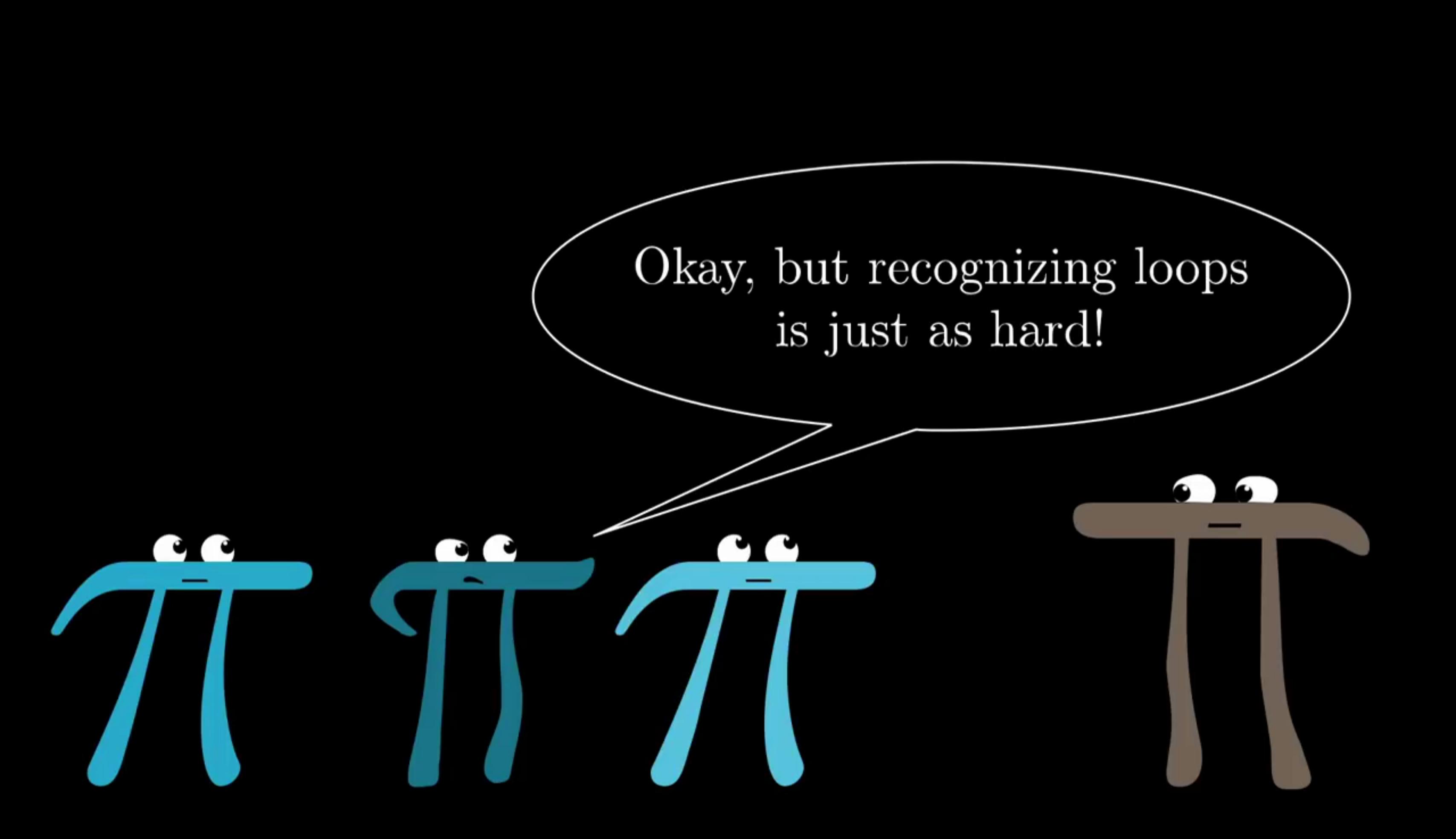


Upper loop neuron...maybe...

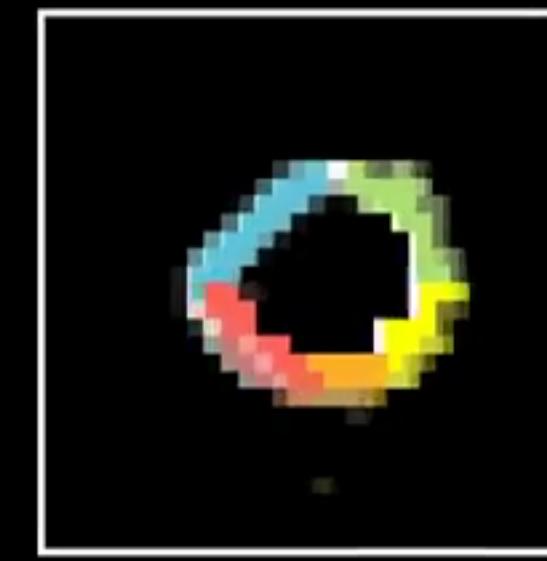
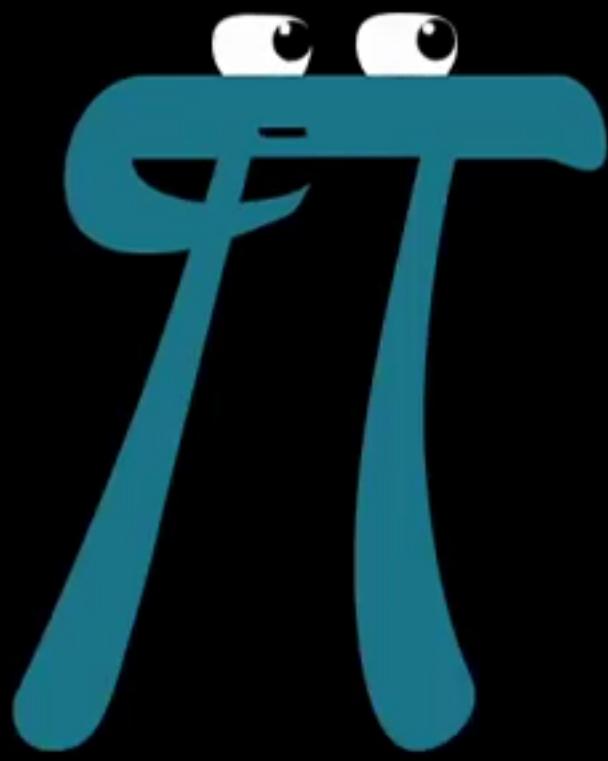


784

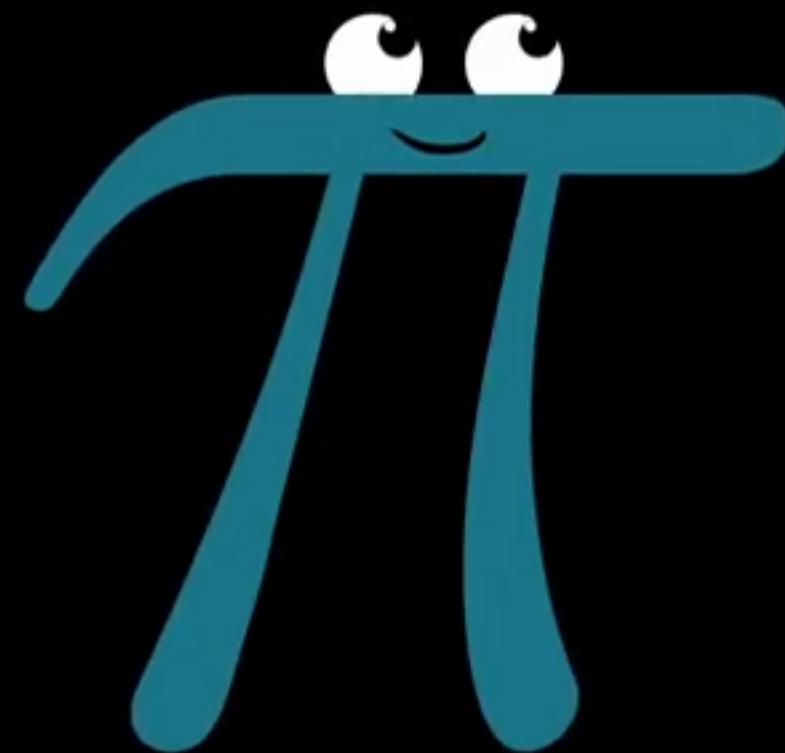




Okay, but recognizing loops
is just as hard!

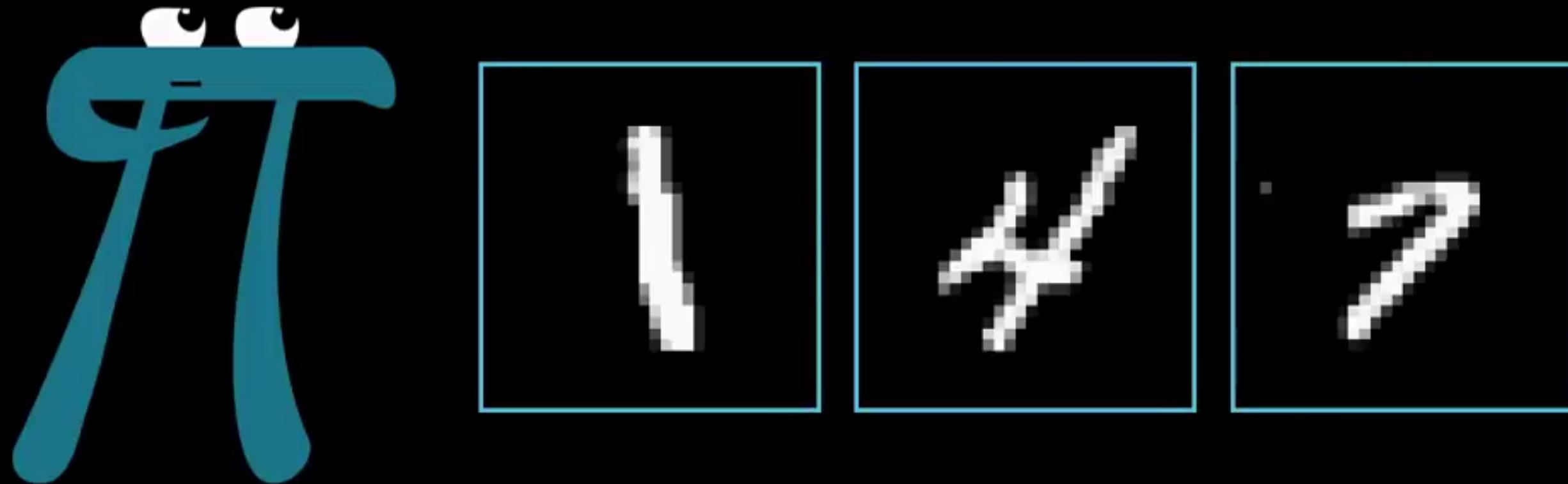


$$\boxed{\text{circle}} = \boxed{\text{cyan shape}} + \boxed{\text{green shape}} + \boxed{\text{yellow shape}} + \boxed{\text{orange shape}} - \boxed{\text{red shape}}$$

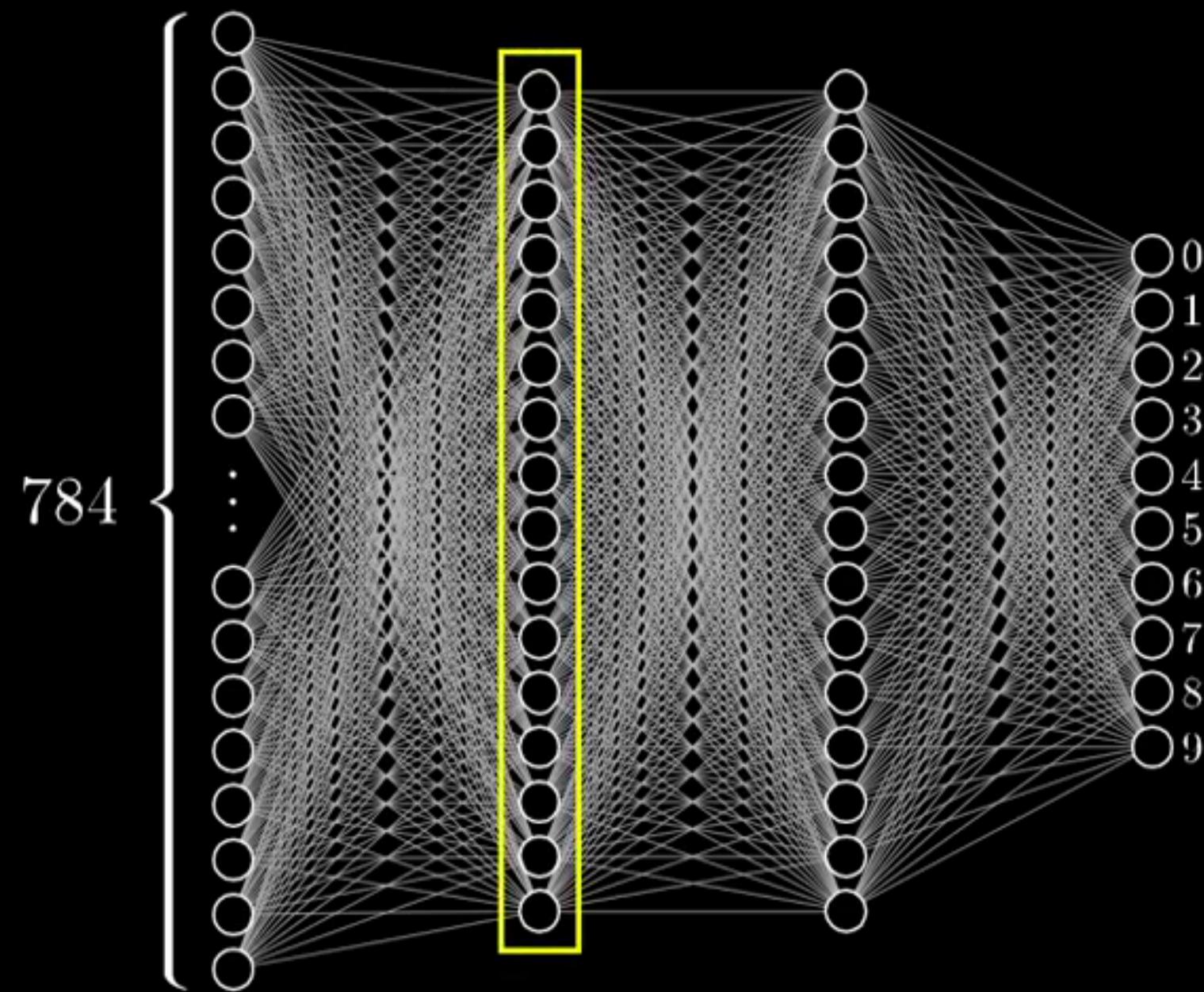


$$\text{apple} = \text{cyan_stem} + \text{green_leaves} + \text{yellow_leaf} + \text{orange_stem} + \text{red_leaf}$$

$$\text{apple} = \text{pink_dot} + \text{purple_dot} + \text{brown_dot}$$

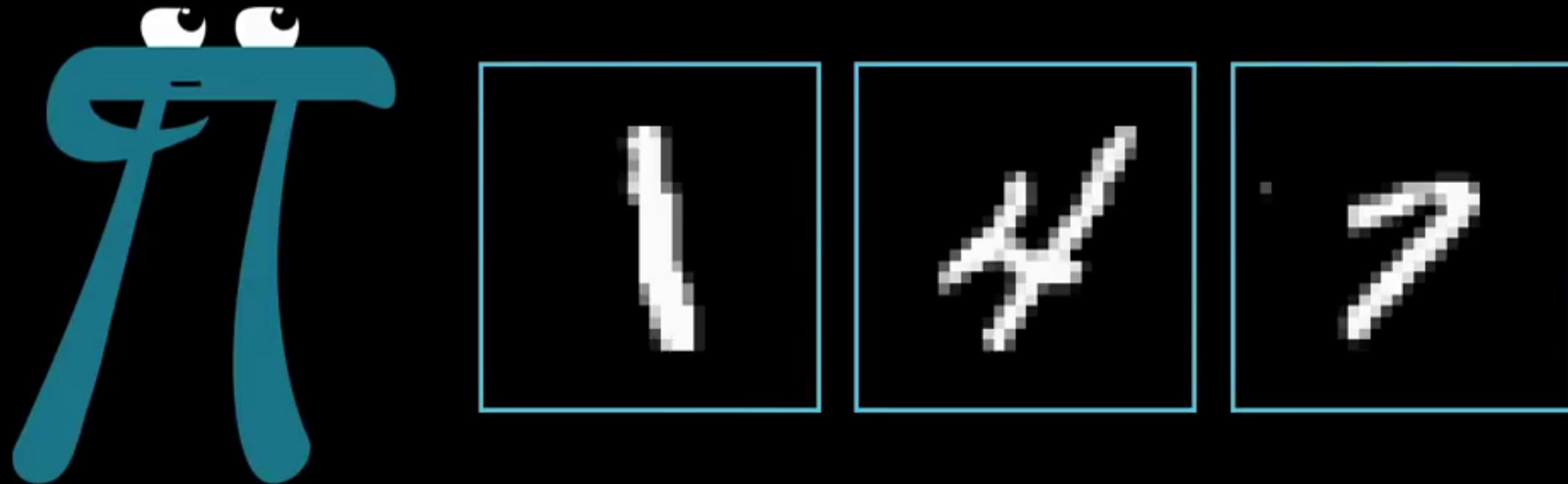


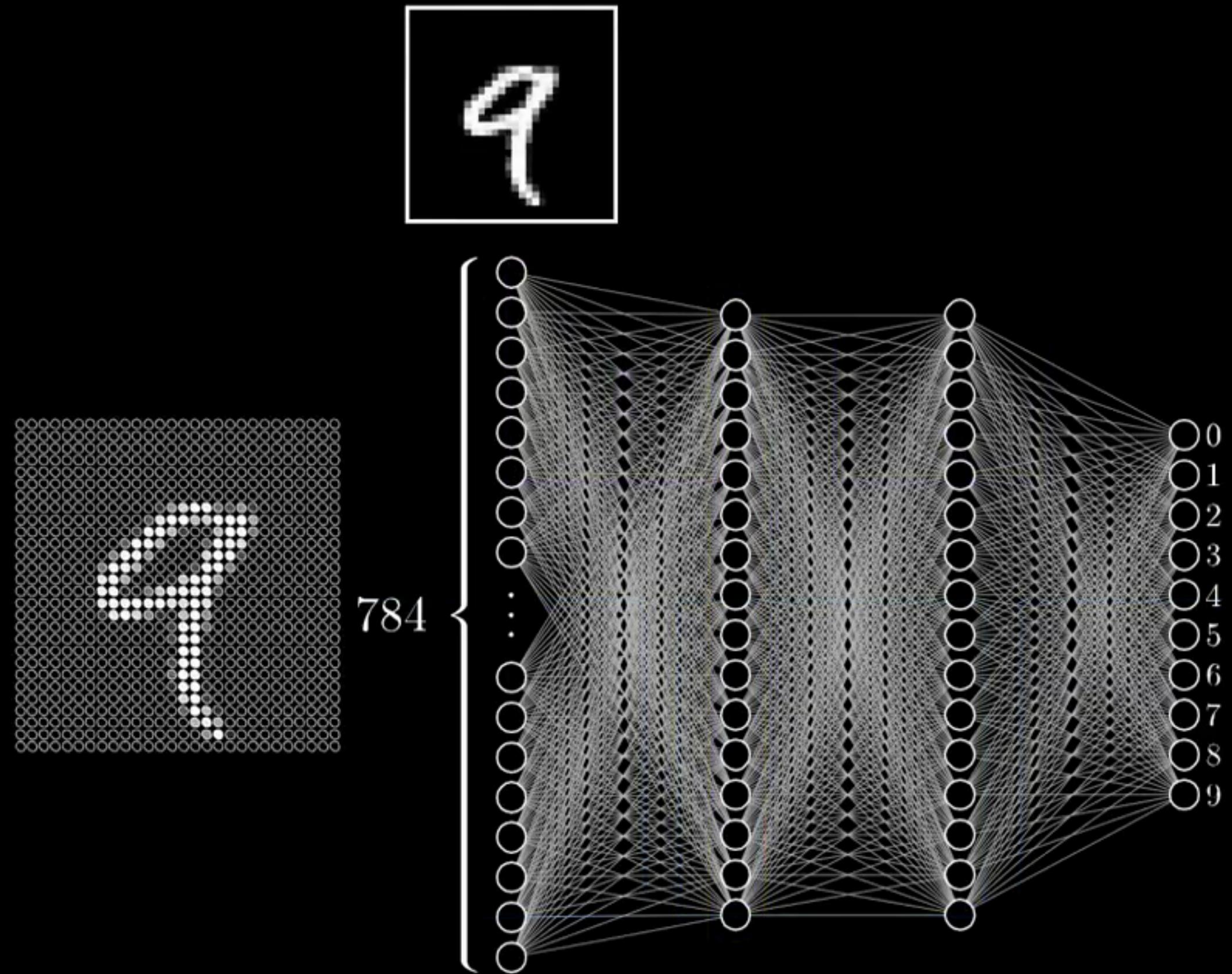
“Little edge” layer?

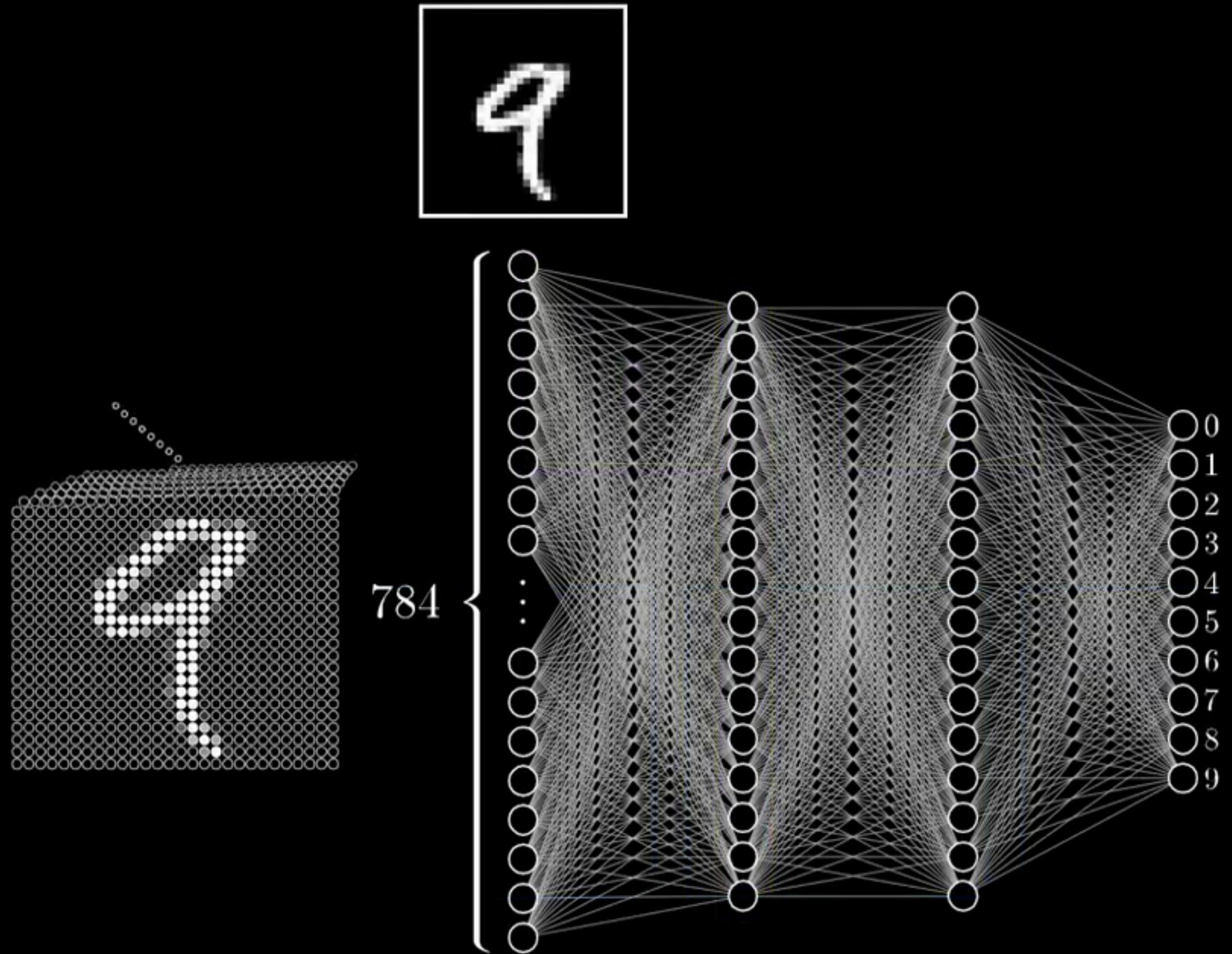


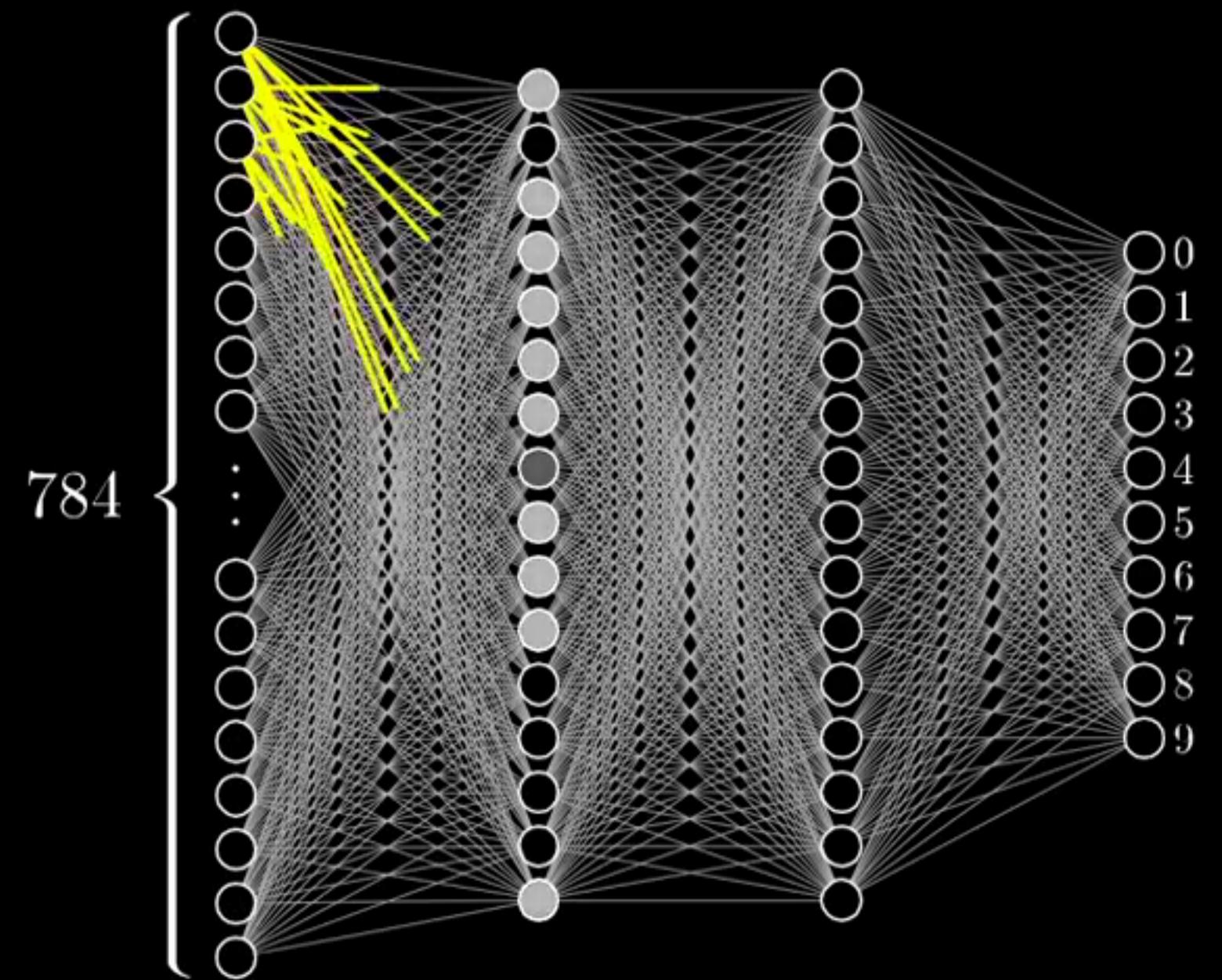
$$\text{apple} = \text{cyan_stem} + \text{green_leaf} + \text{yellow_leaf} + \text{orange_stem} + \text{red_leaf}$$

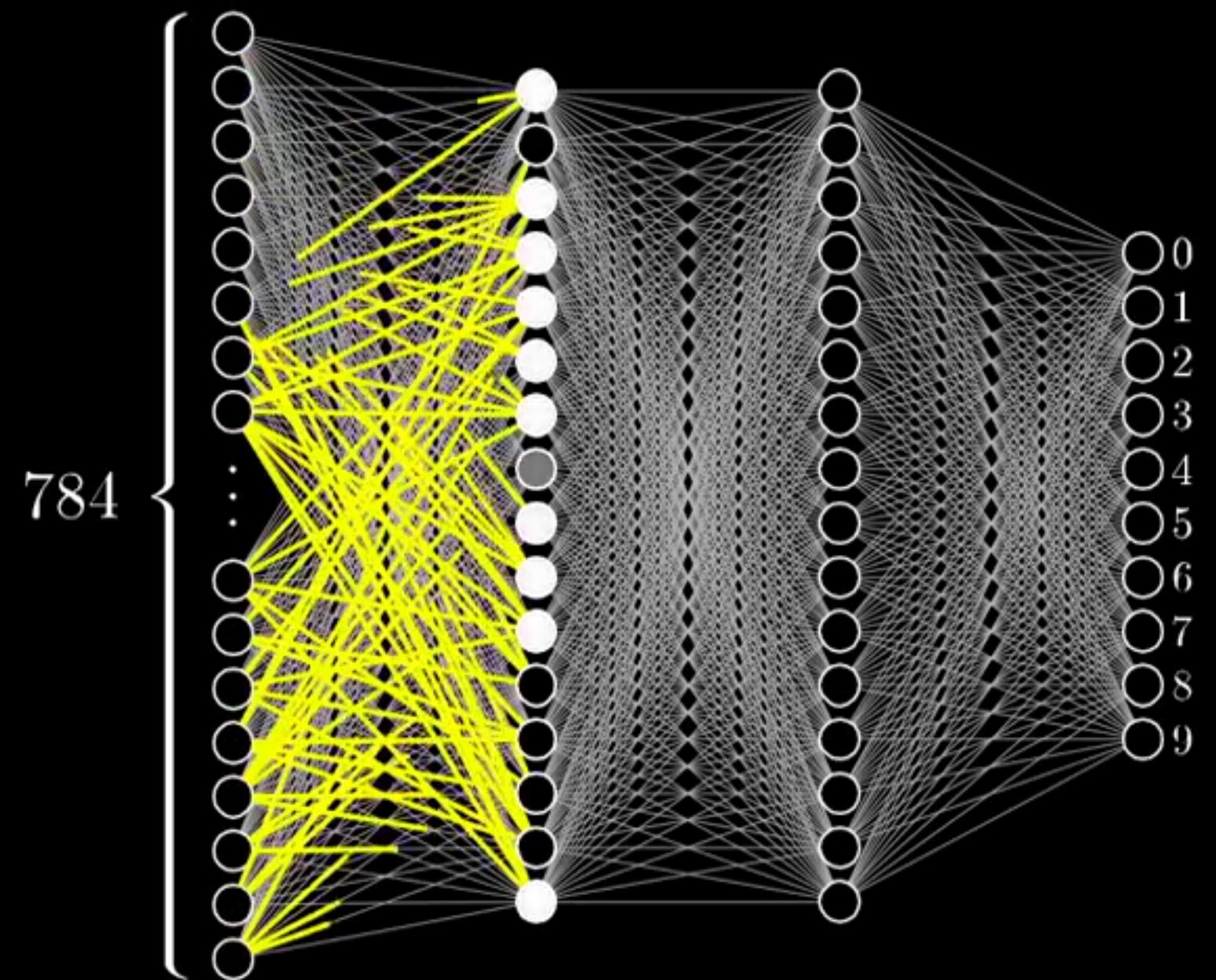
$$\text{apple} = \text{pink_dot} + \text{purple_dot} + \text{brown_dot}$$

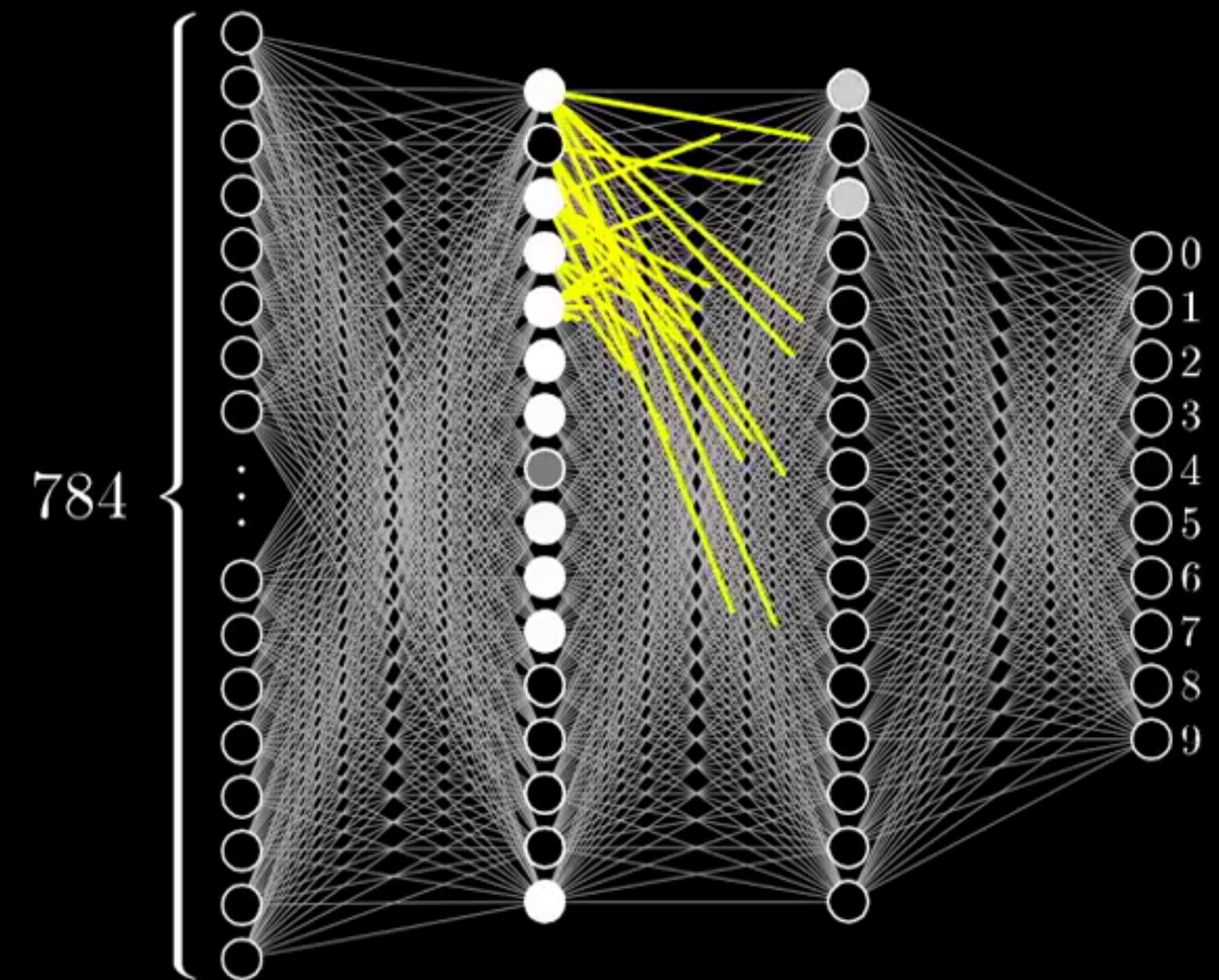


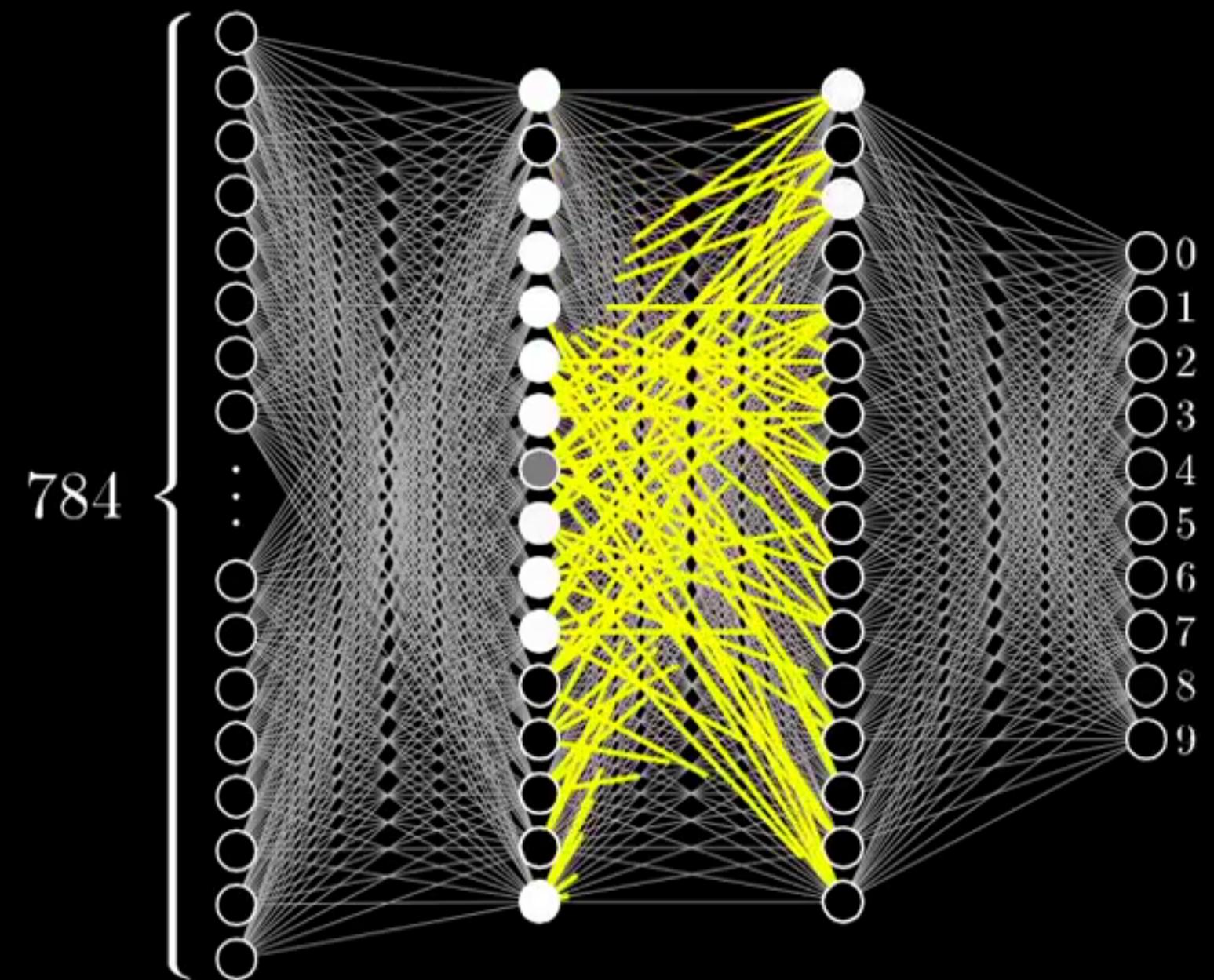








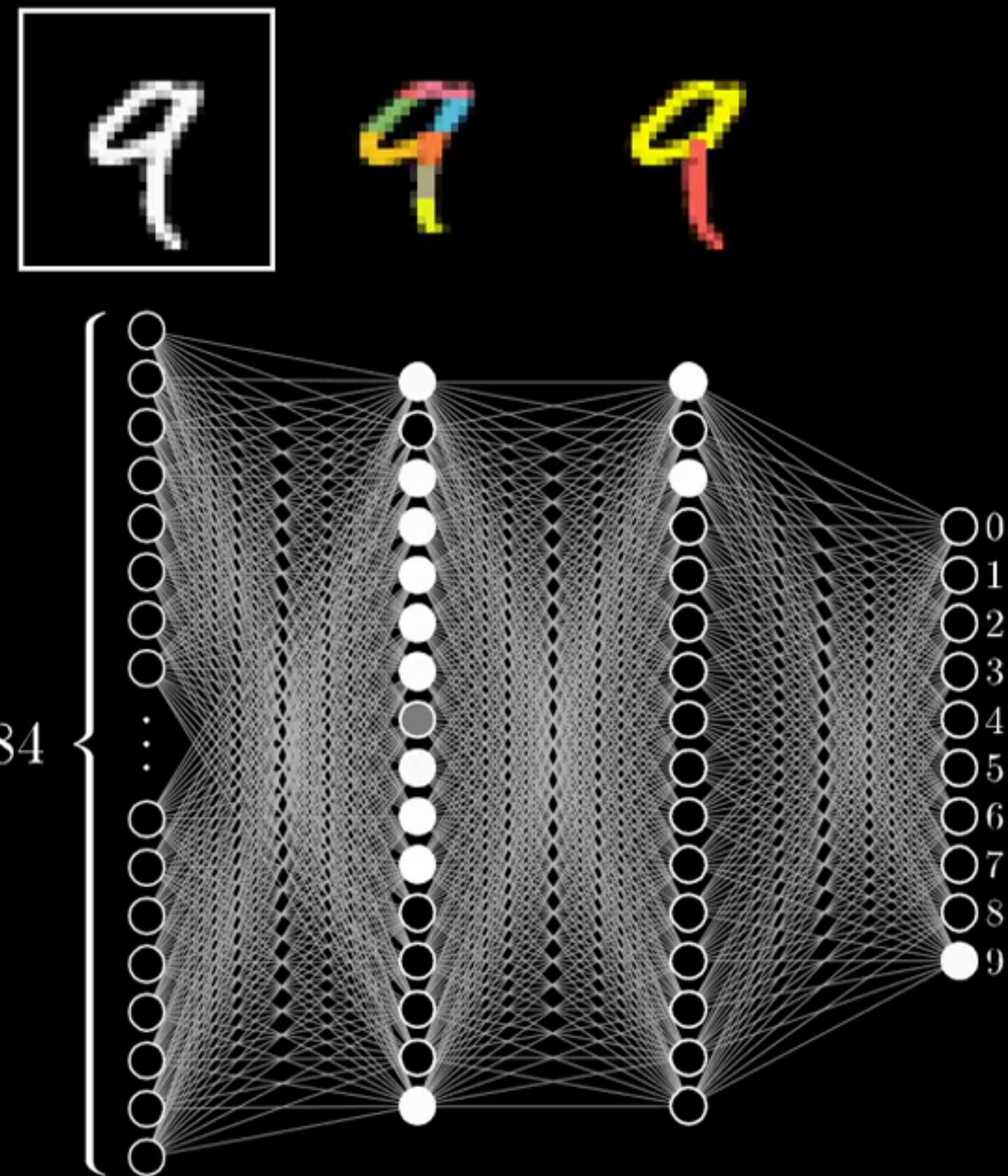




We'll get back
to this



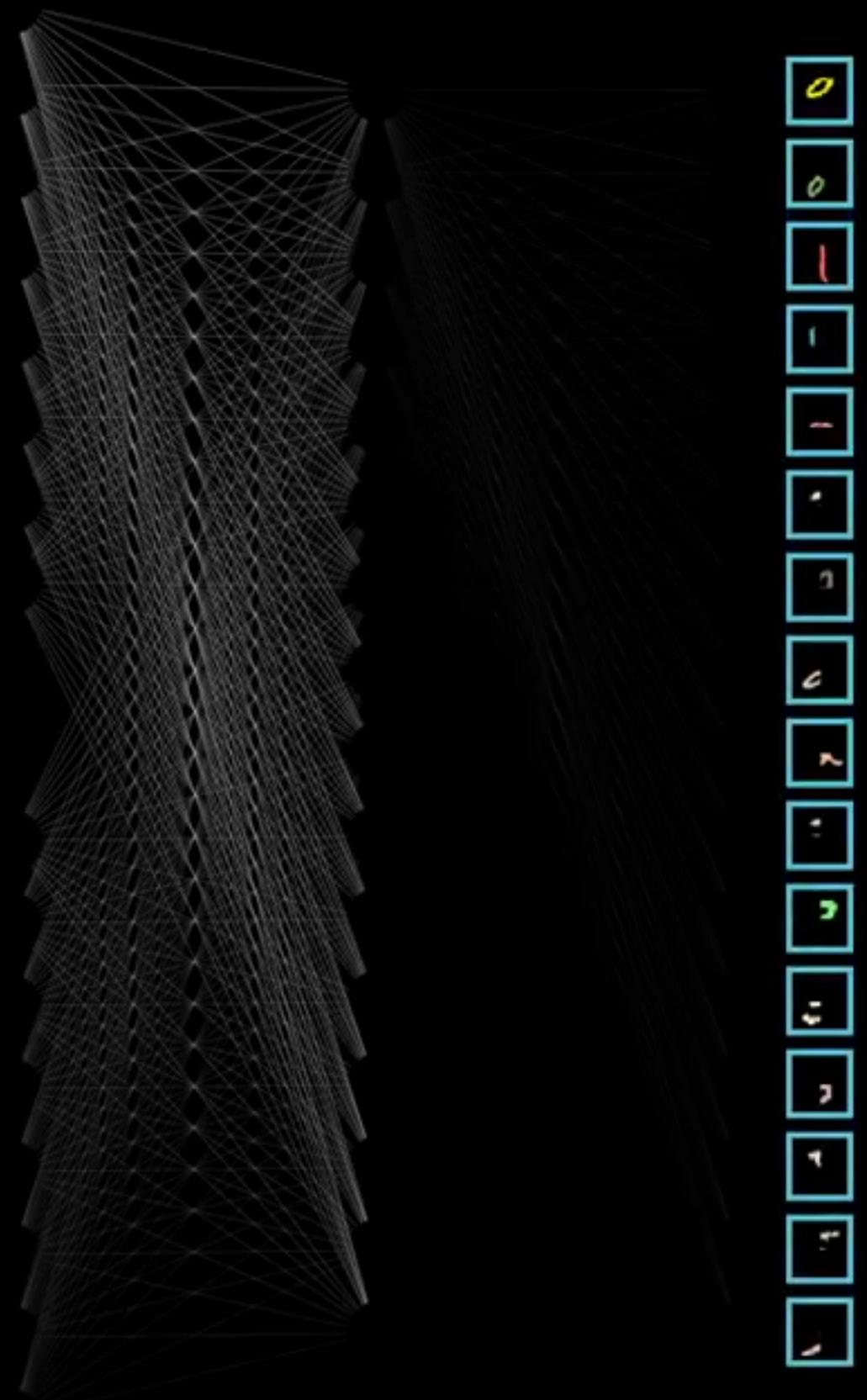
Does the network
actually do this?



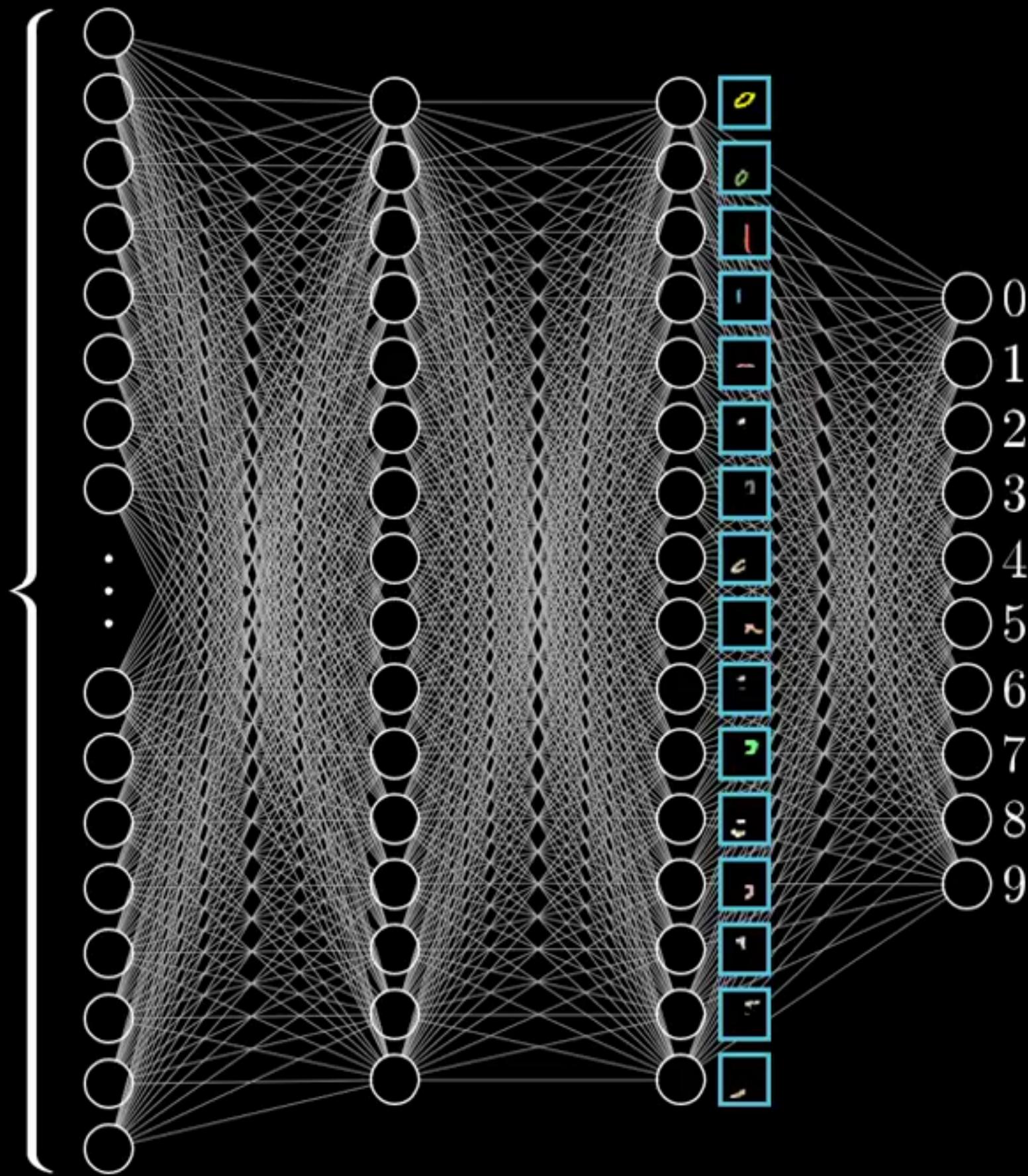
$$9 = \text{yellow loop} + \text{red vertical stroke}$$

$$8 = \text{yellow loop} + \text{green circle}$$

$$4 = \text{red vertical stroke} + \text{cyan vertical stroke} + \text{pink horizontal stroke}$$



784









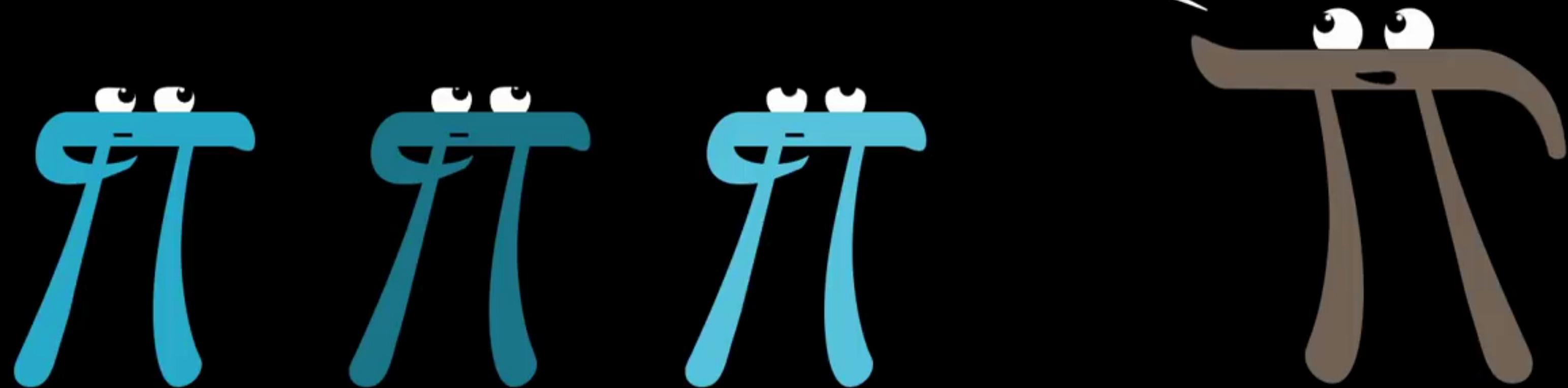


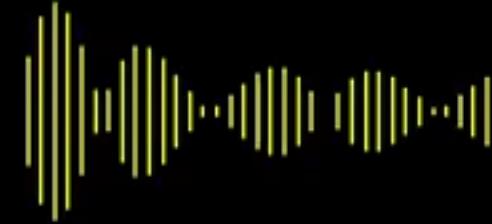


Raw audio



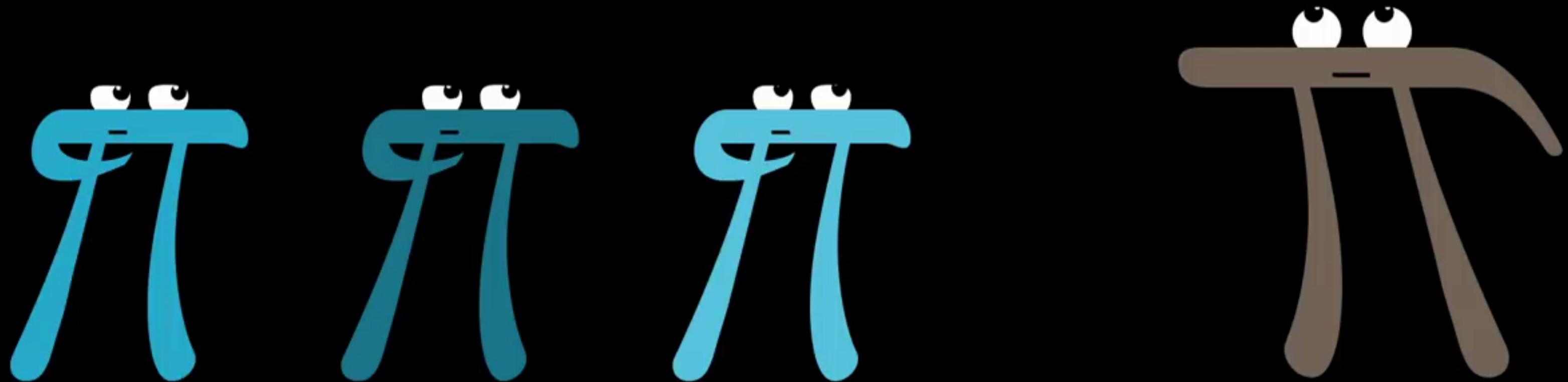
recognition



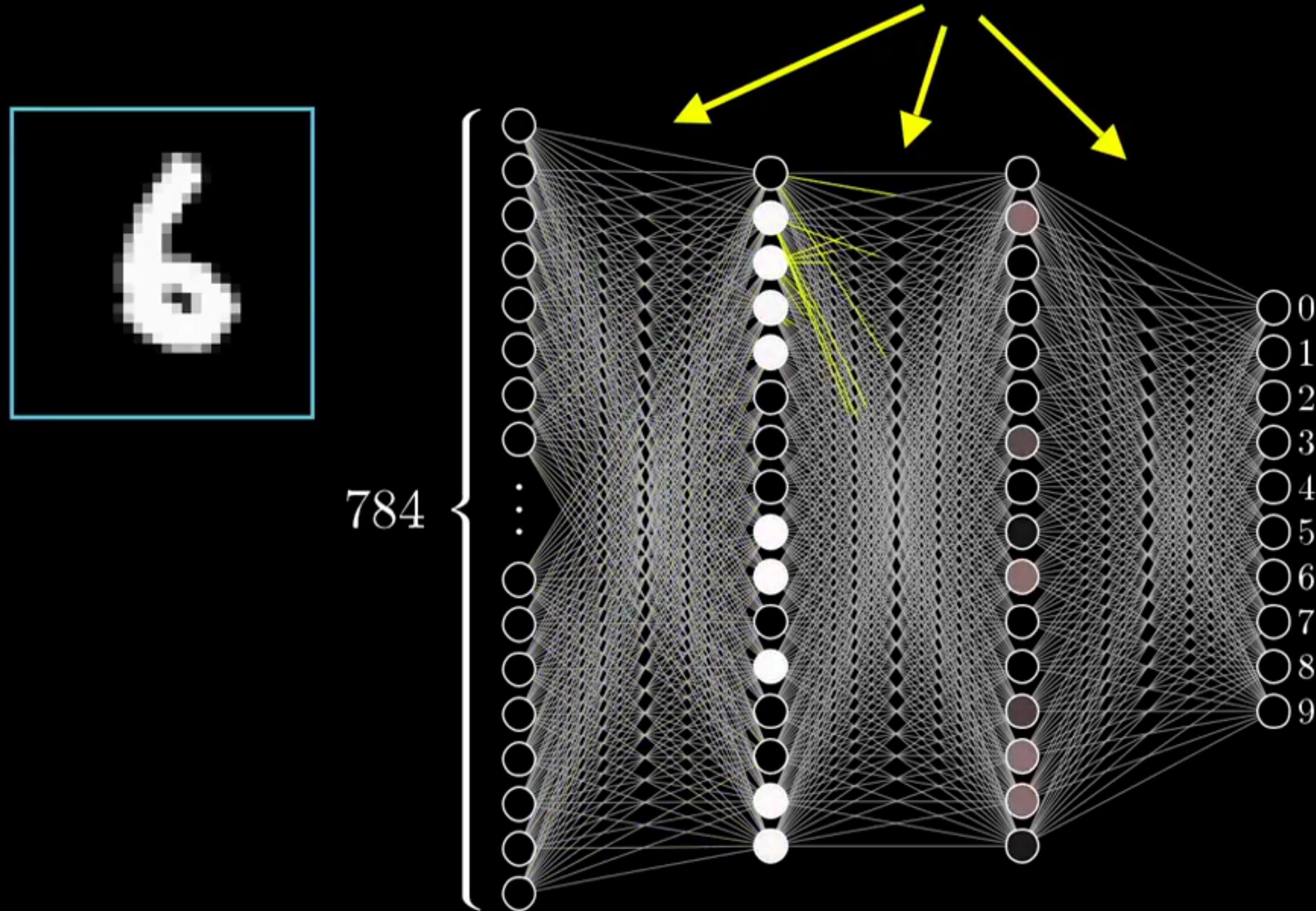


Raw audio → recognition → re·cog·ni·tion → recognition

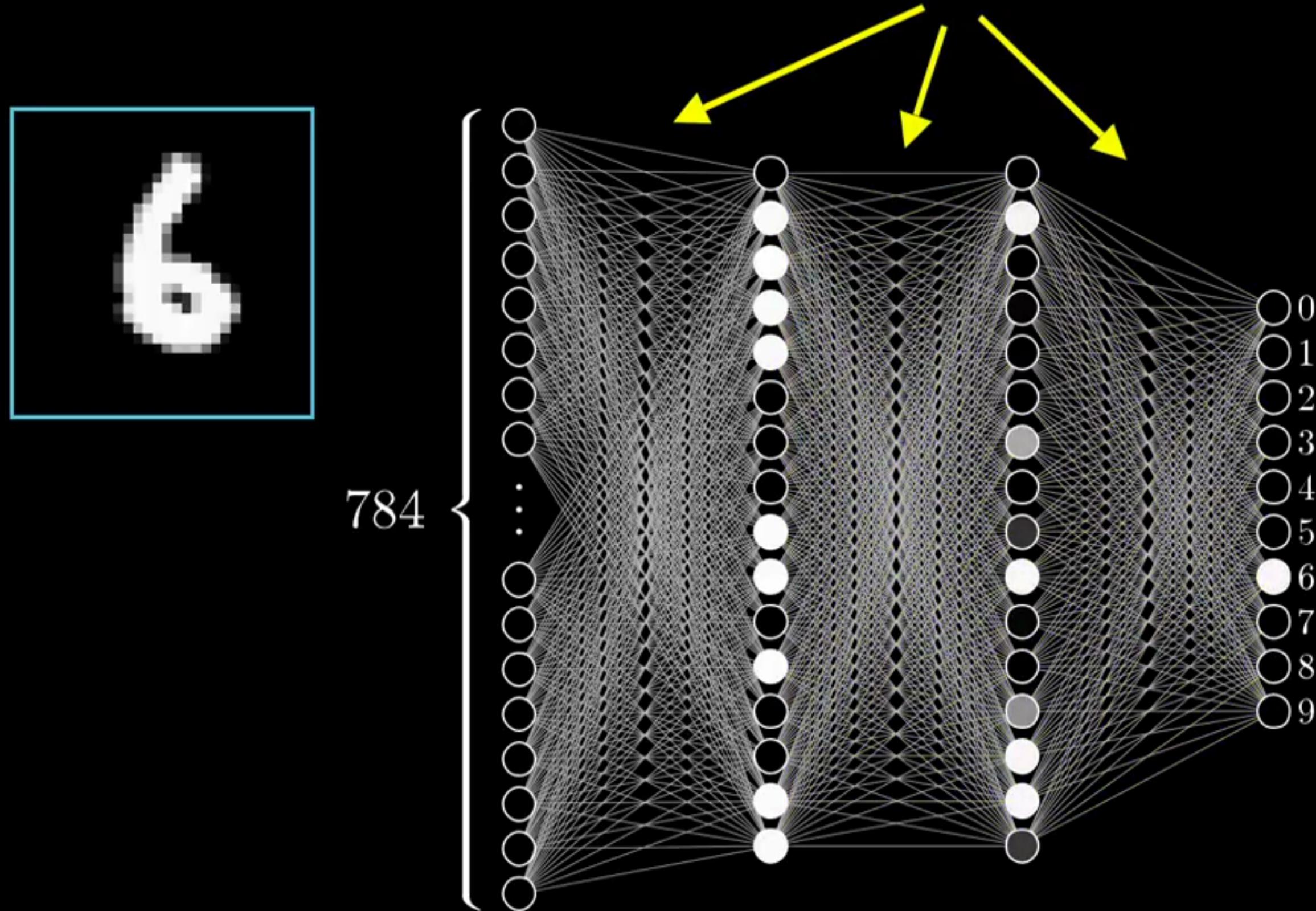
Raw audio

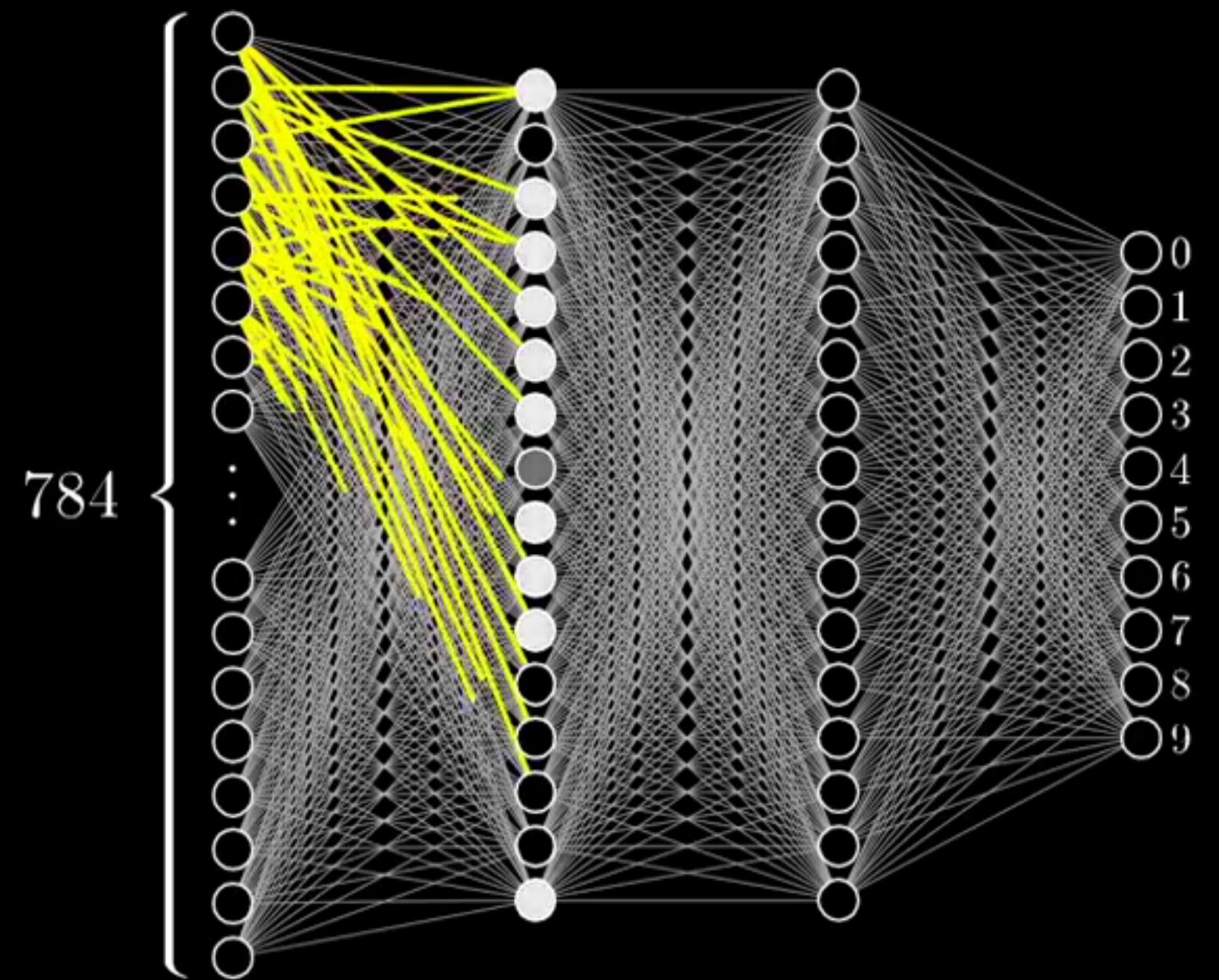


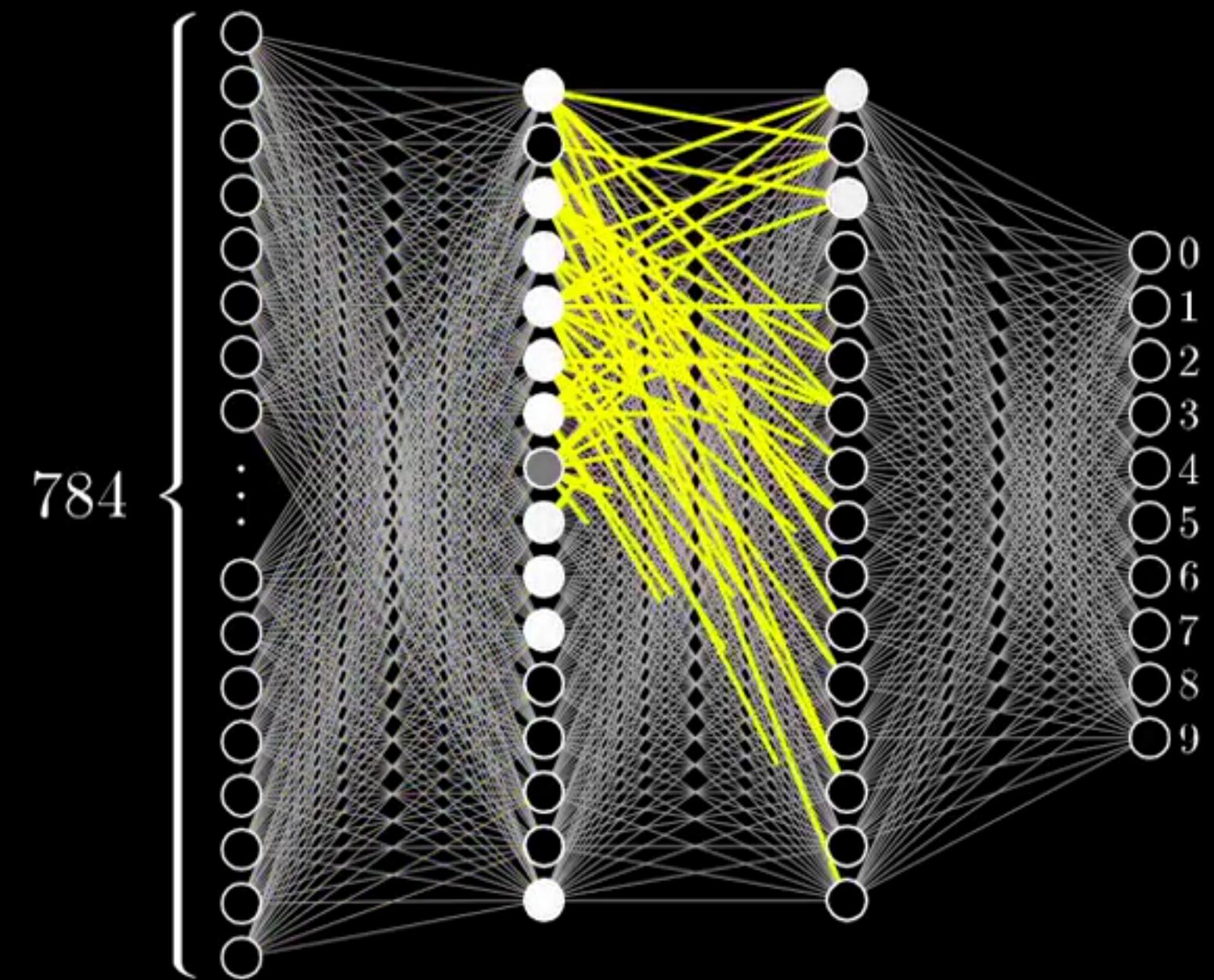
What are these connections actually doing?

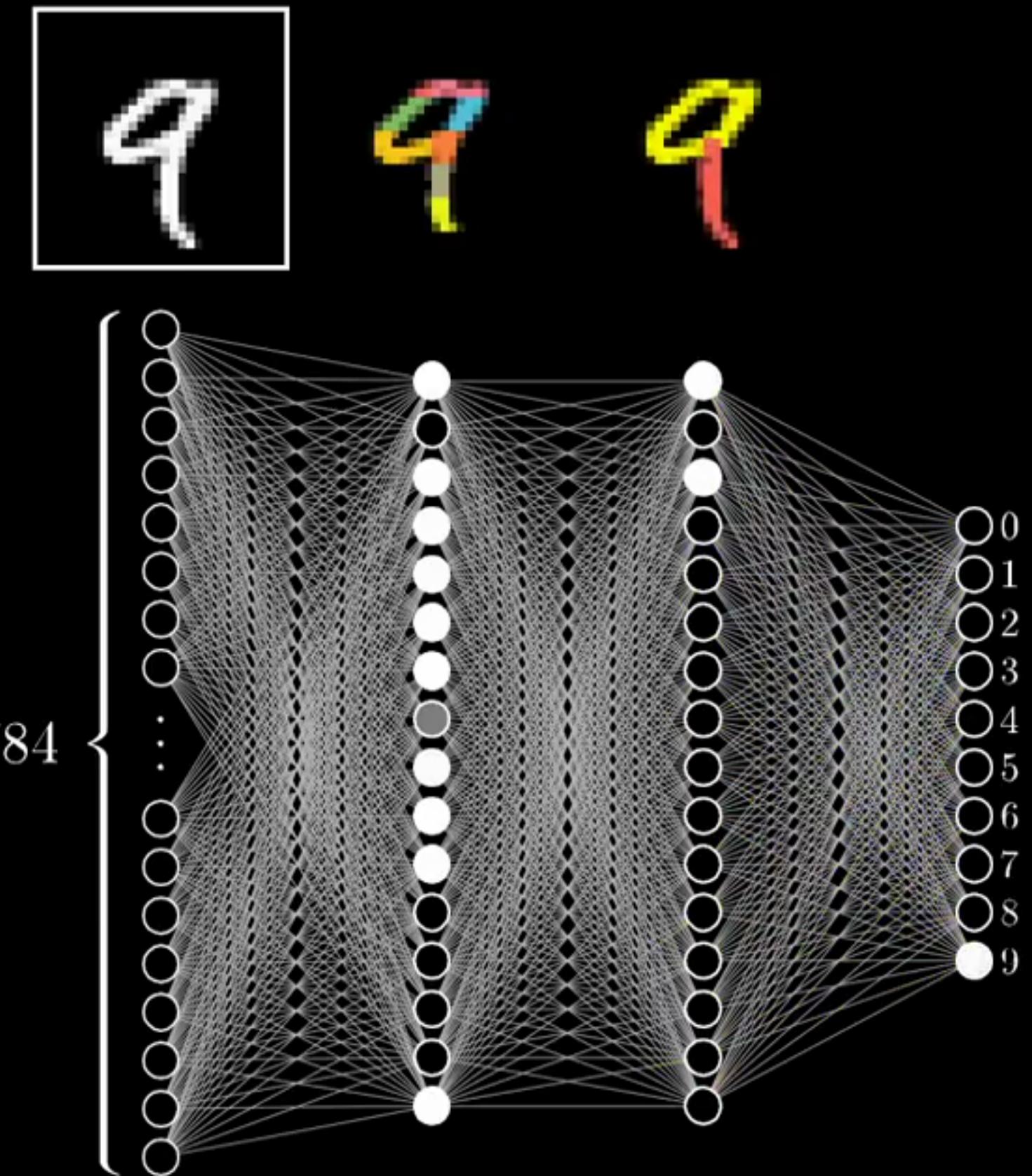


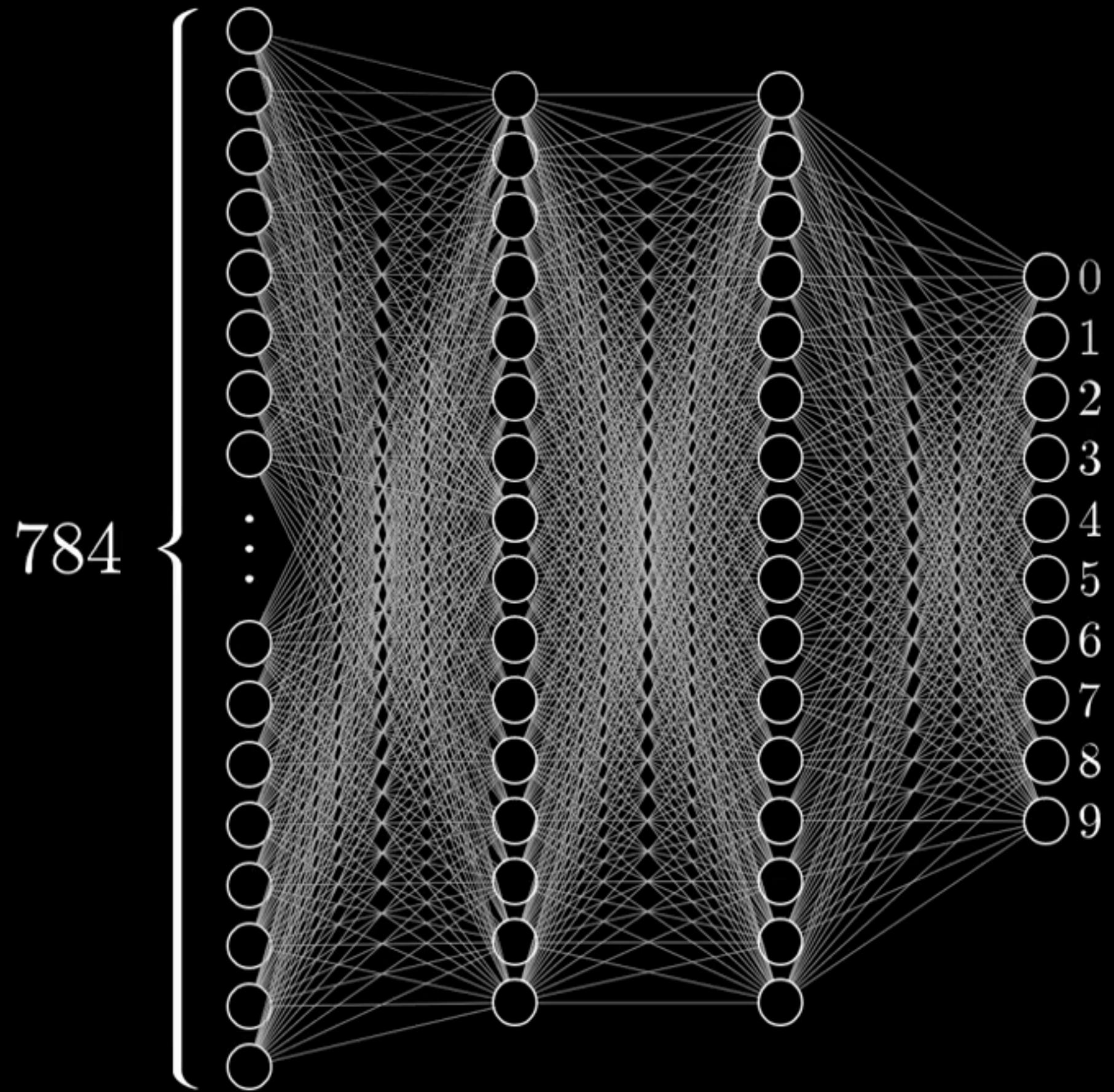
What are these connections actually doing?





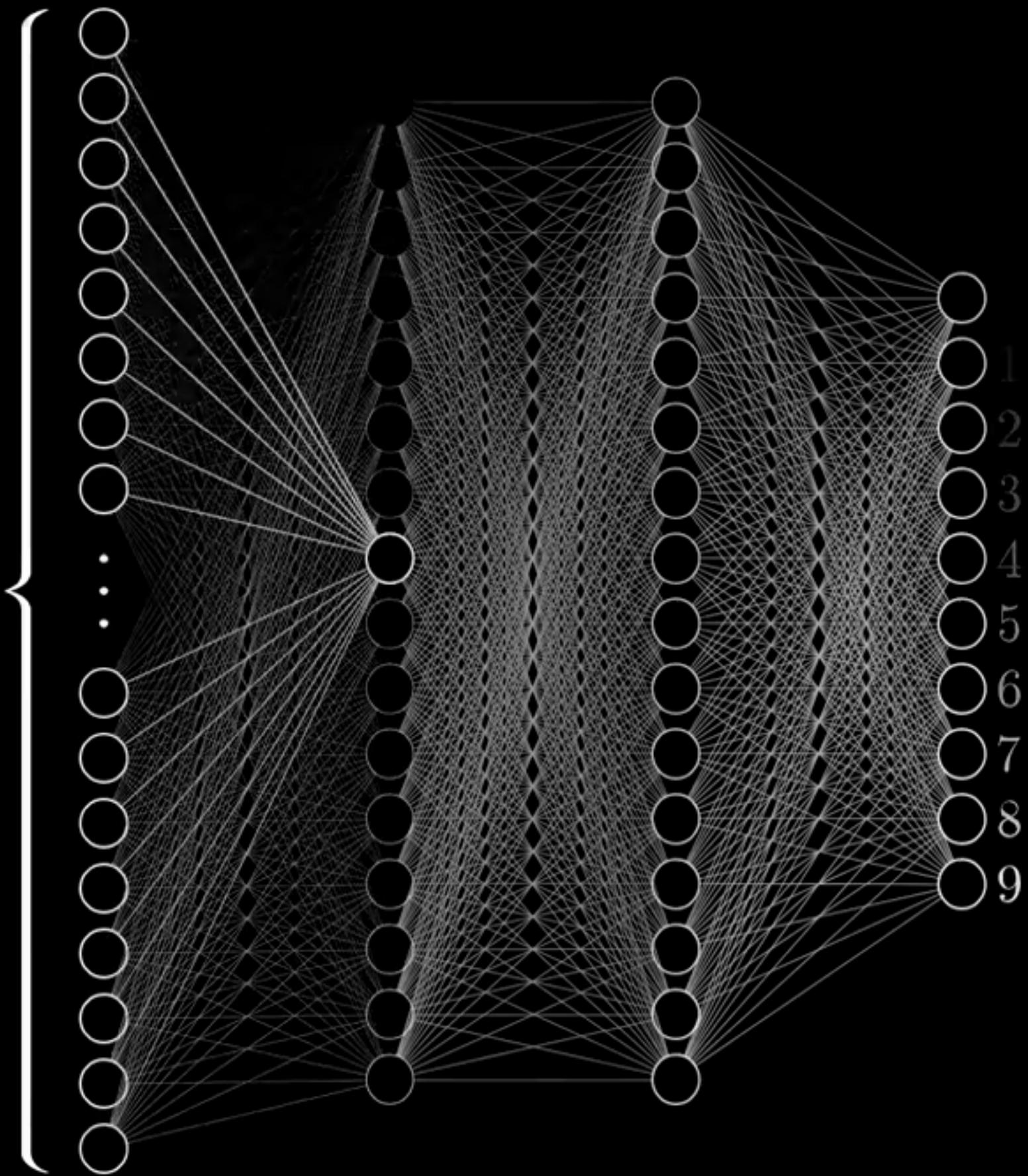


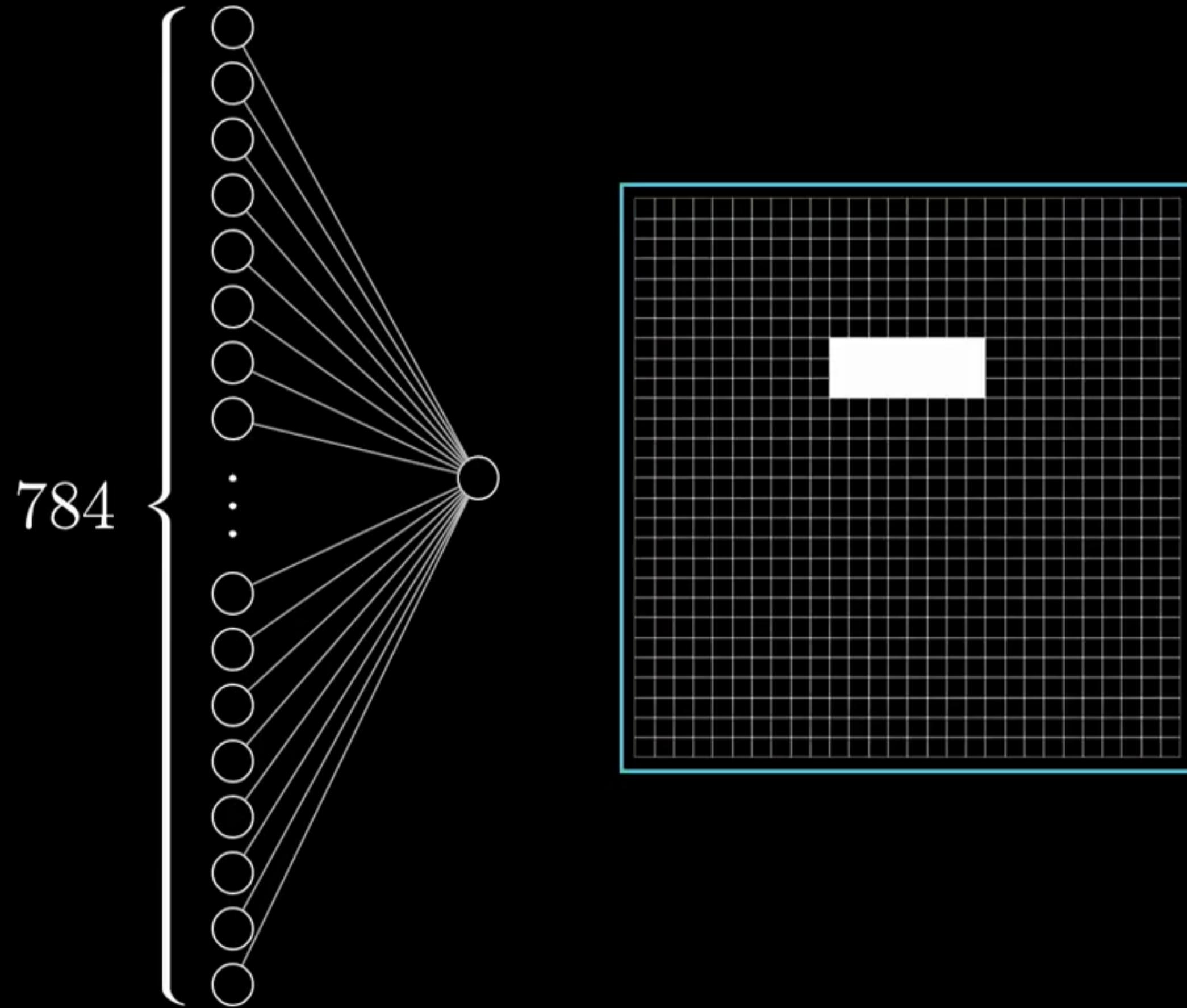




0
1
2
3
4
5
6
7
8
9

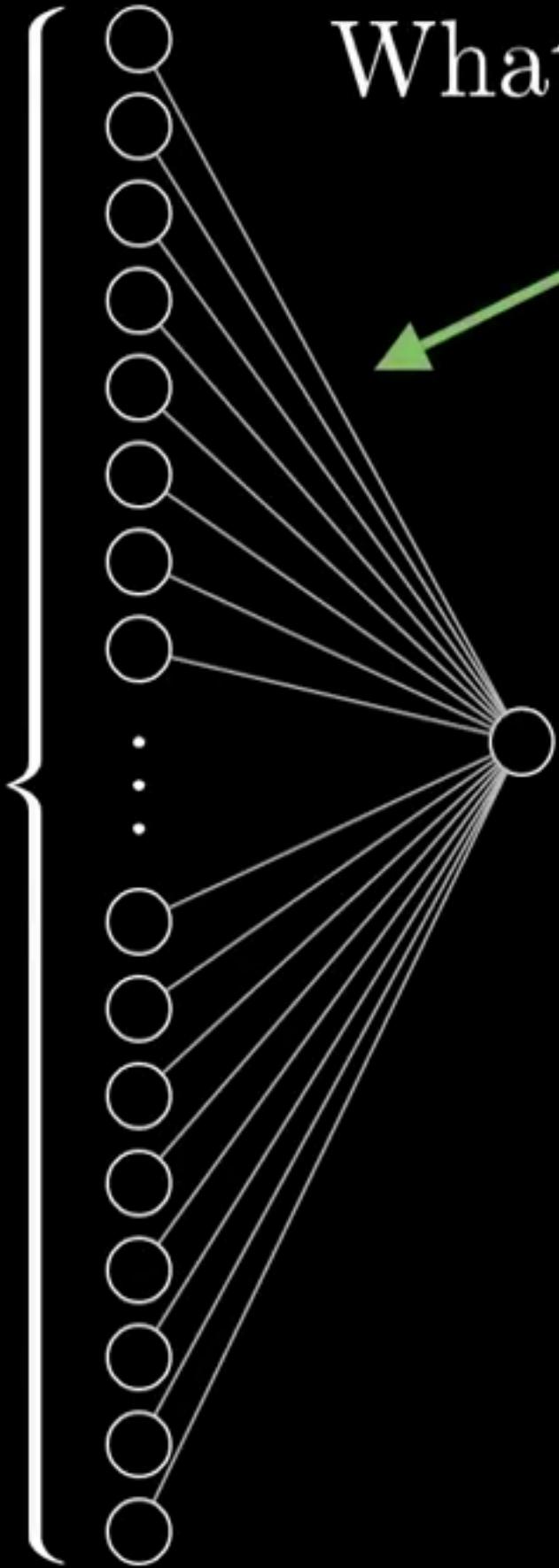
784



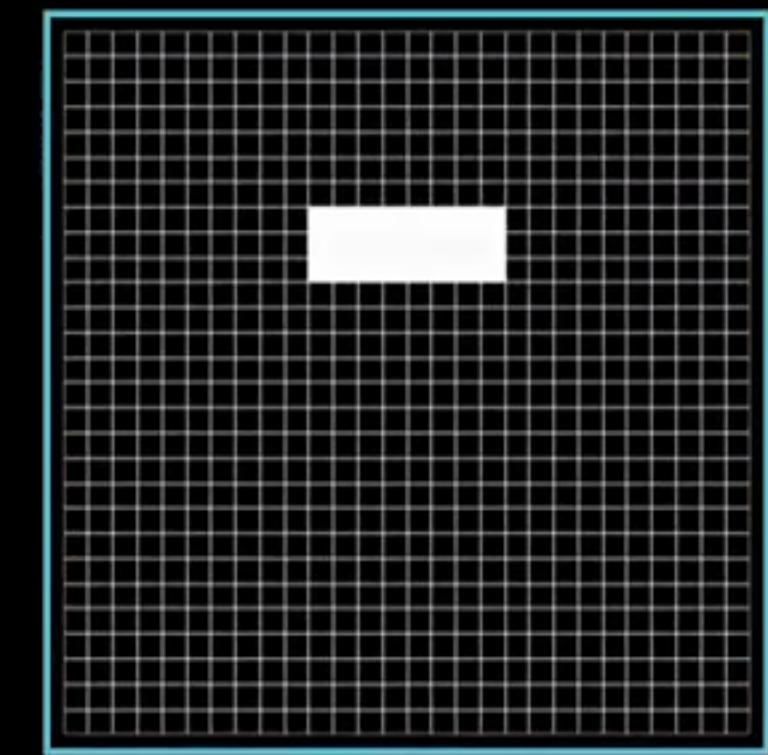


What parameters should exist?

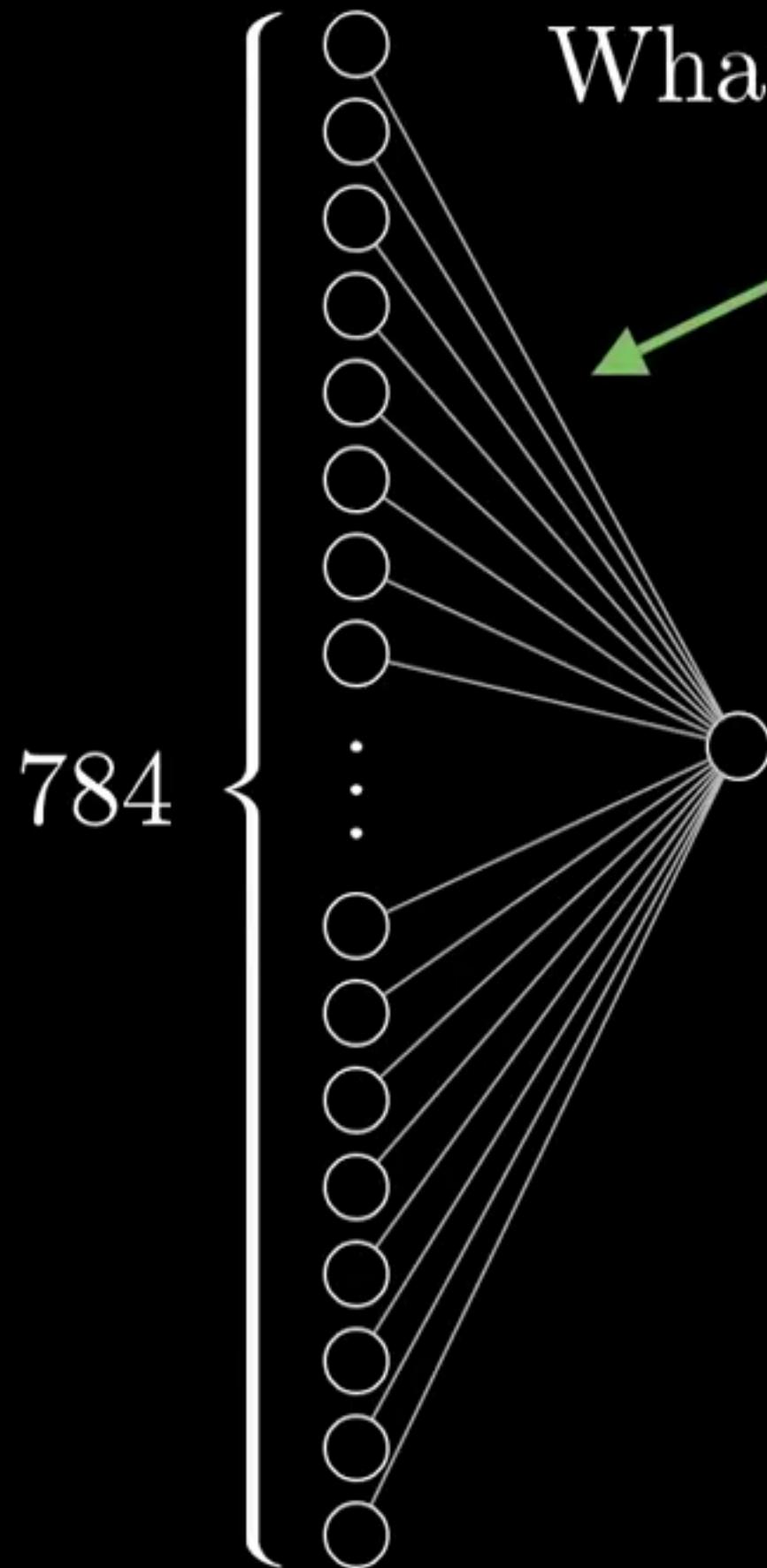
784



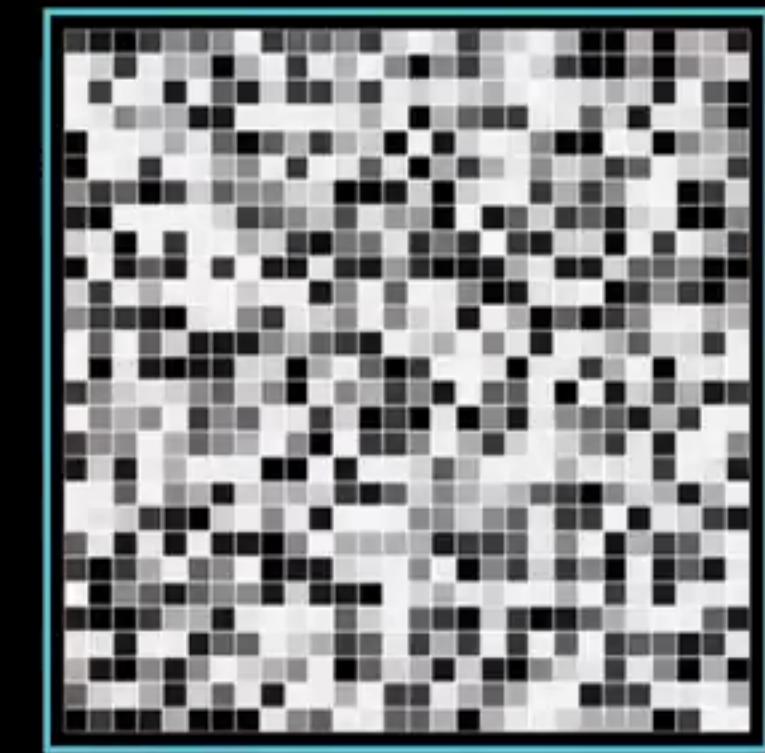
- $p_1: 0.00$
- $p_2: 0.00$
- $p_3: 0.00$
- $p_4: 0.00$
- $p_5: 0.00$
- $p_6: 0.00$
- $p_7: 0.00$
- $p_8: 0.00$
- \vdots



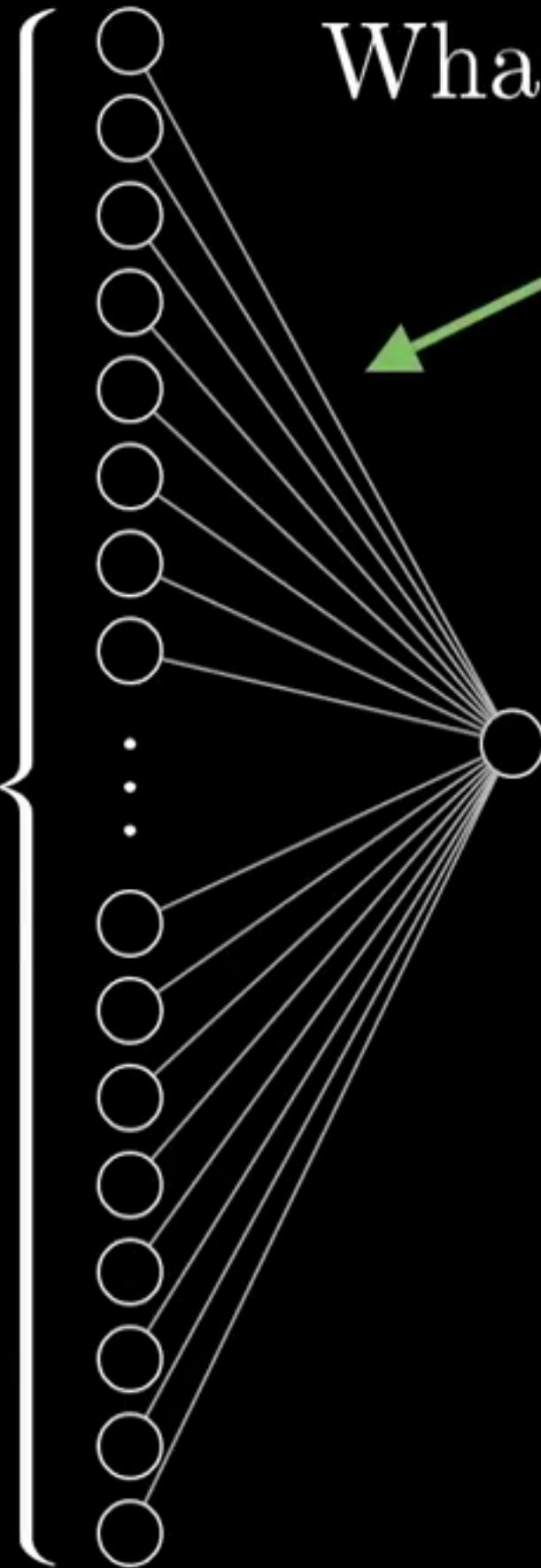
What parameters should exist?



- $p_1: 0.03$
- $p_2: 0.85$
- $p_3: 0.39$
- $p_4: 1.00$
- $p_5: 0.99$
- $p_6: 0.71$
- $p_7: 0.39$
- $p_8: 0.11$
- \vdots

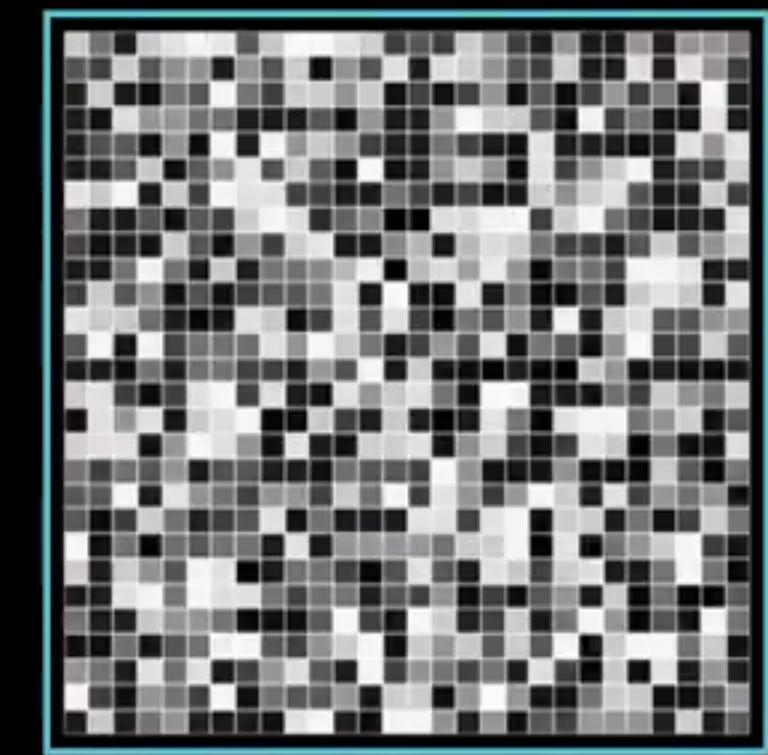


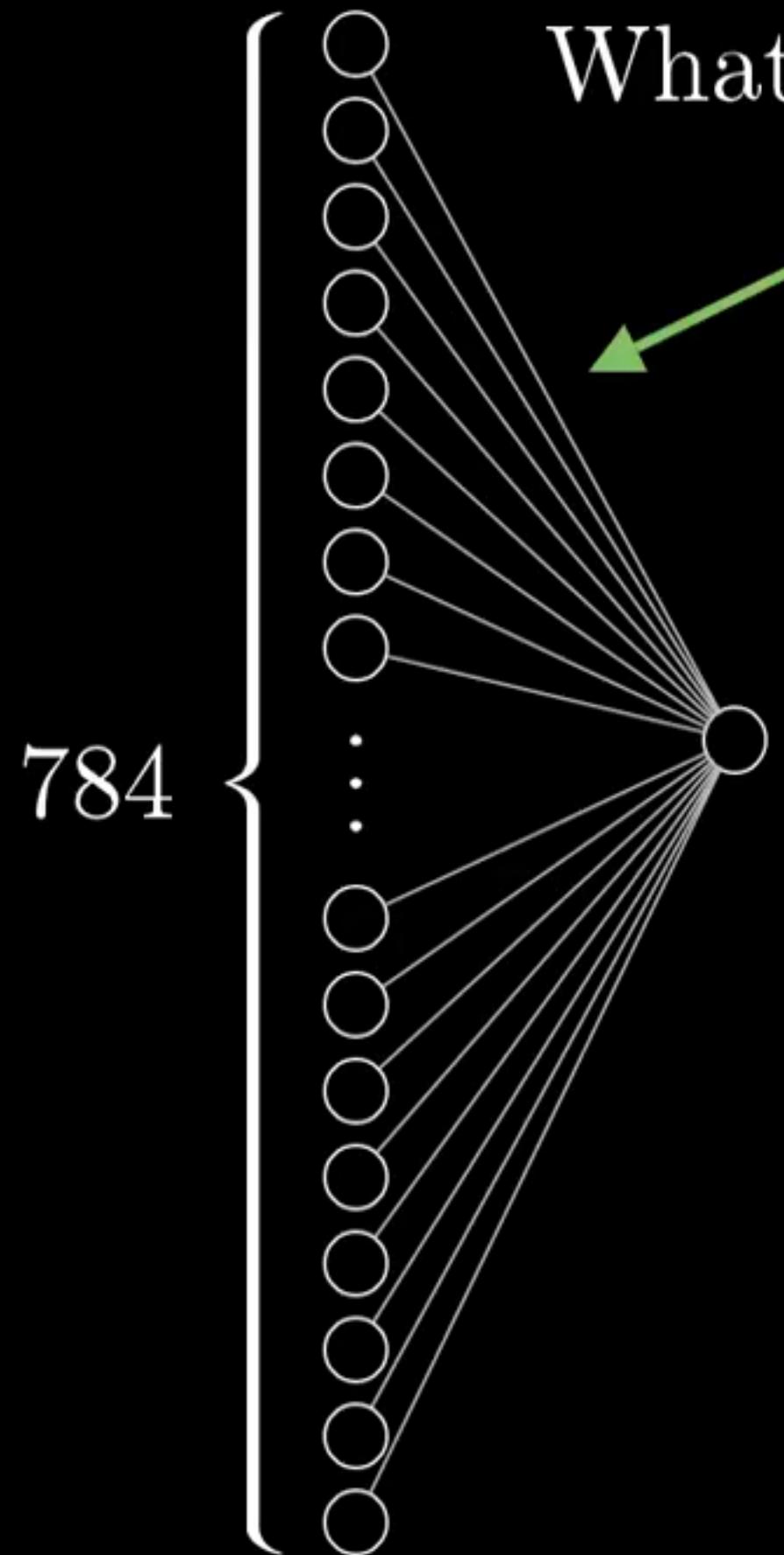
784



What parameters should exist?

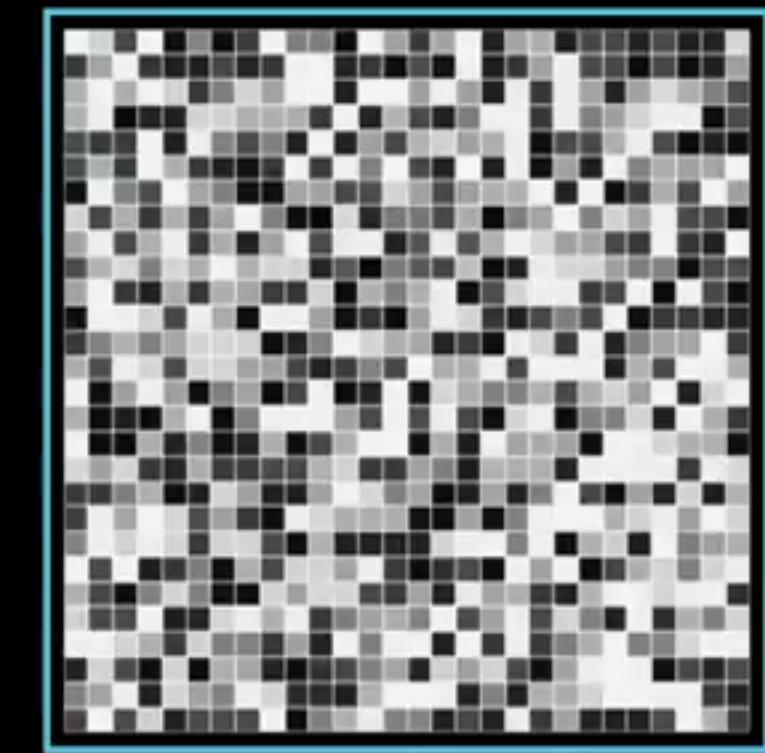
- $p_1: 0.14$
- $p_2: 0.23$
- $p_3: 1.00$
- $p_4: 0.06$
- $p_5: 0.35$
- $p_6: 0.59$
- $p_7: 1.00$
- $p_8: 0.47$
- \vdots





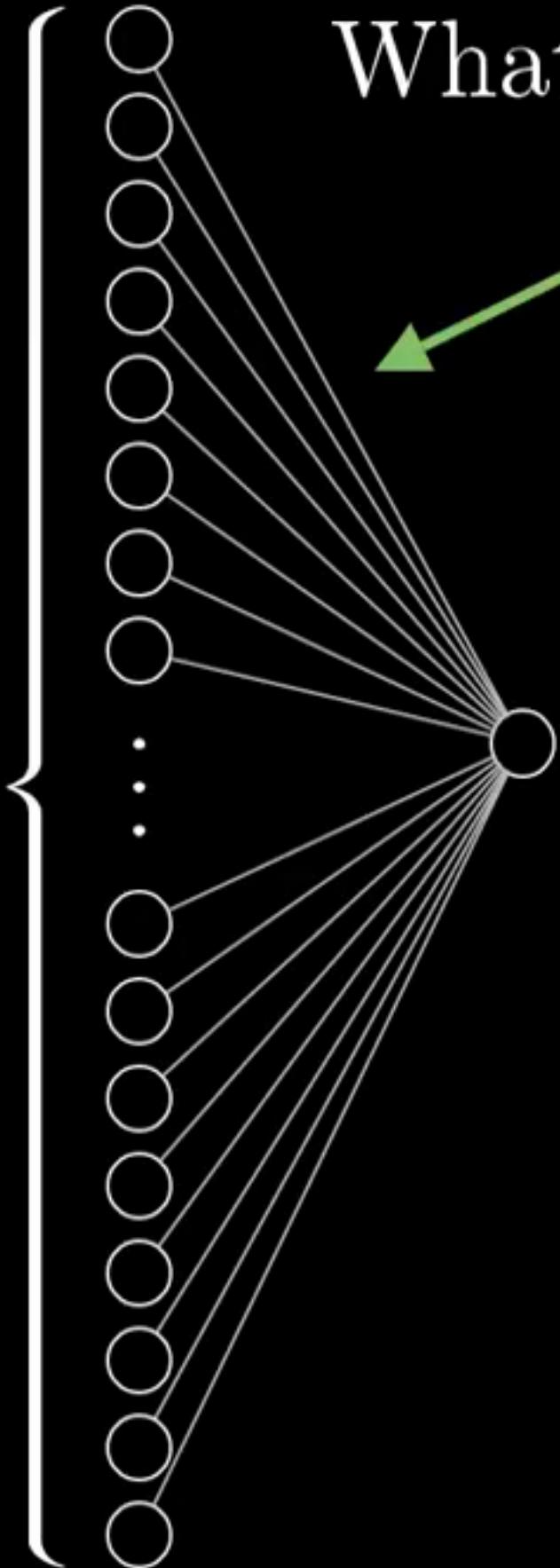
What parameters should exist?

- $p_1: 0.31$
- $p_2: 0.67$
- $p_3: 0.51$
- $p_4: 0.71$
- $p_5: 0.17$
- $p_6: 0.14$
- $p_7: 0.51$
- $p_8: 0.85$
- \vdots

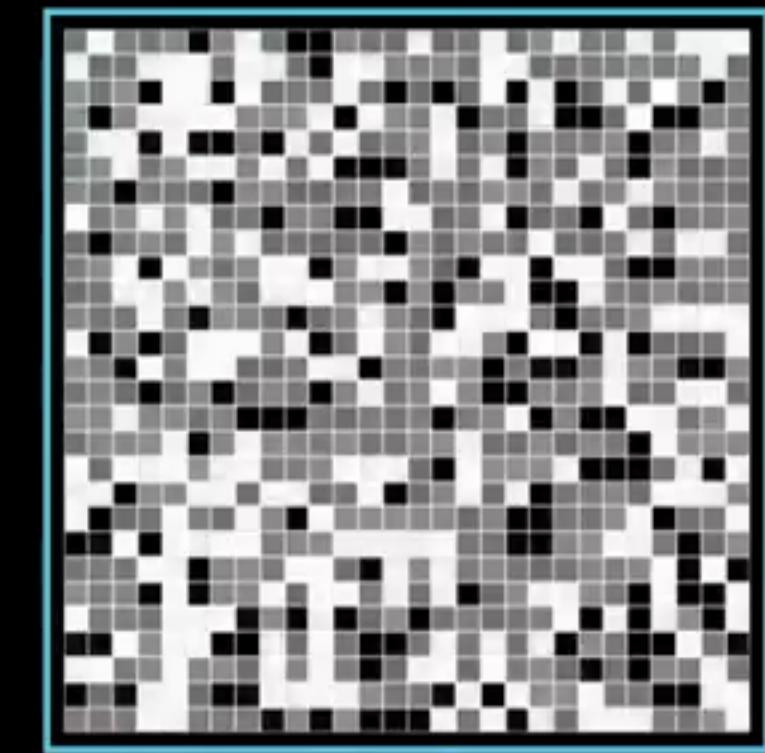


What parameters should exist?

784

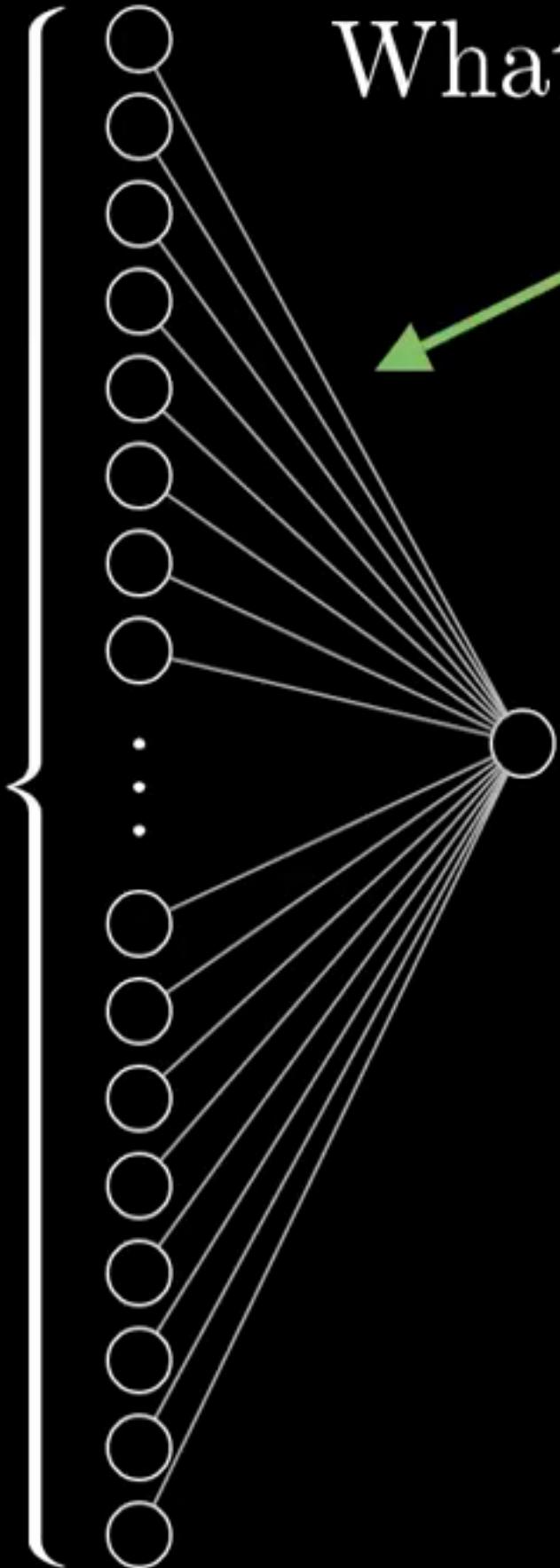


- $p_1: 0.51$
- $p_2: 0.43$
- $p_3: 0.00$
- $p_4: 0.59$
- $p_5: 0.99$
- $p_6: 1.00$
- $p_7: 0.00$
- $p_8: 1.00$
- \vdots

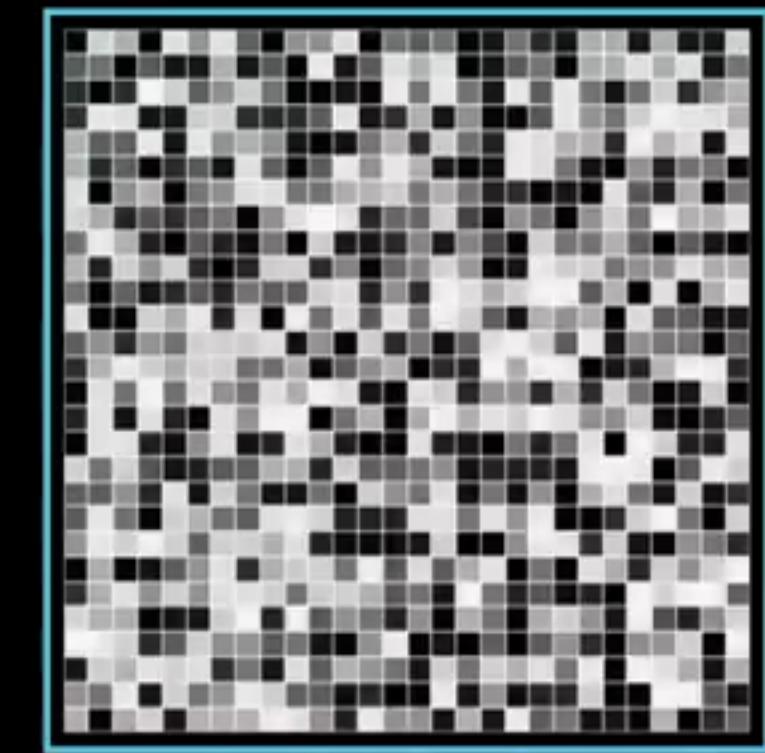


What parameters should exist?

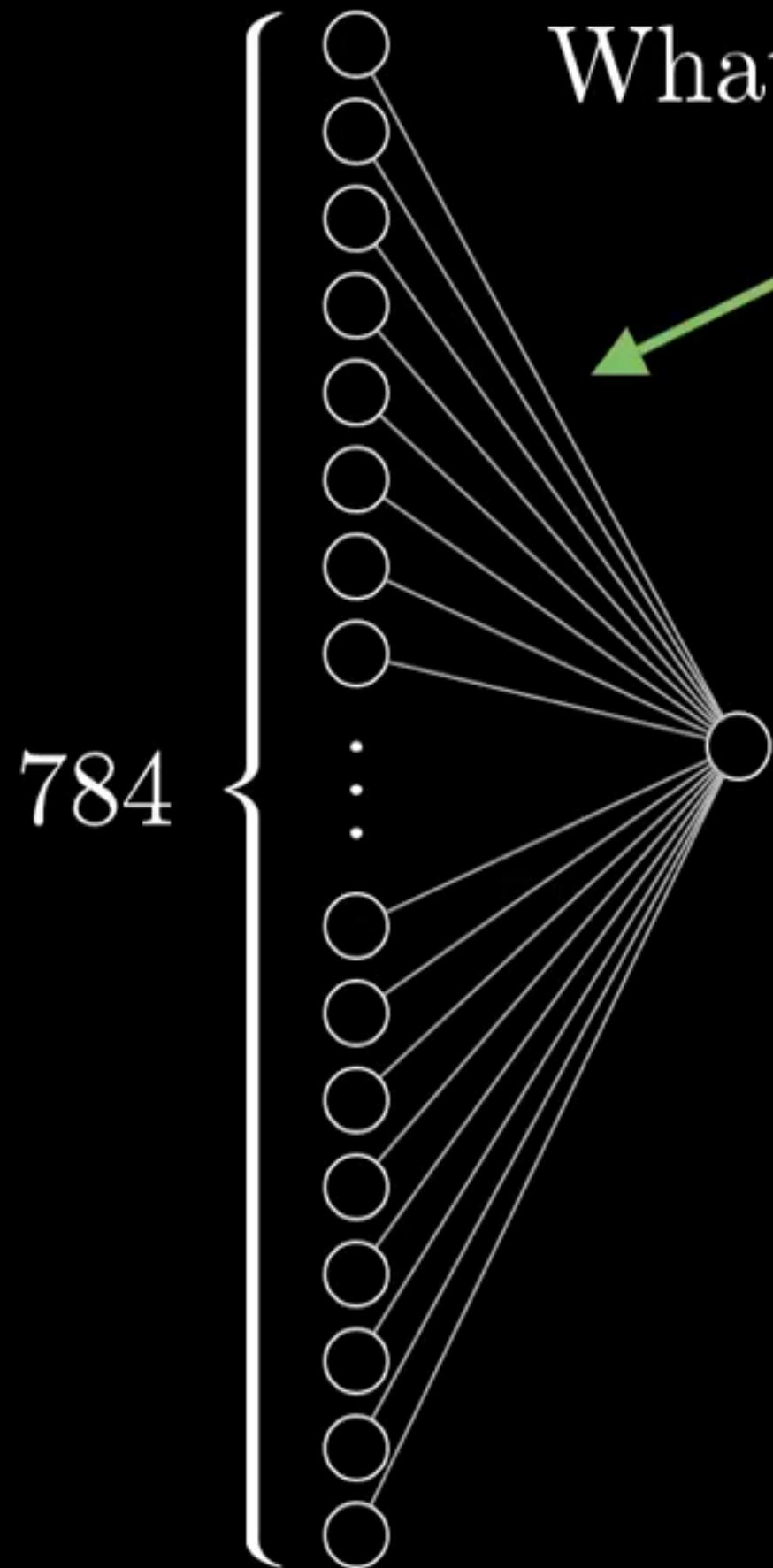
784



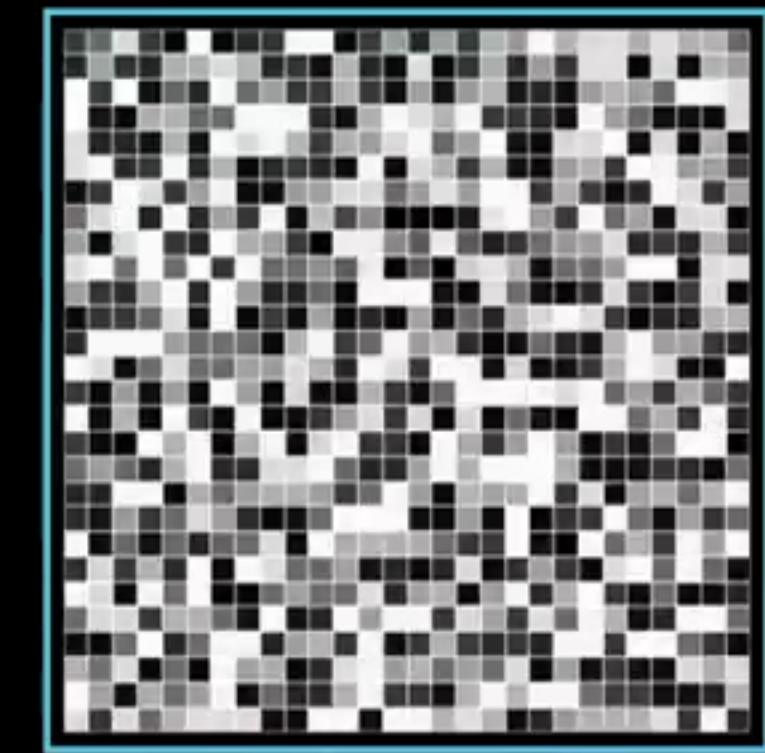
- $p_1: 0.71$
- $p_2: 0.47$
- $p_3: 0.59$
- $p_4: 0.14$
- $p_5: 0.35$
- $p_6: 0.06$
- $p_7: 0.59$
- $p_8: 0.82$
- \vdots



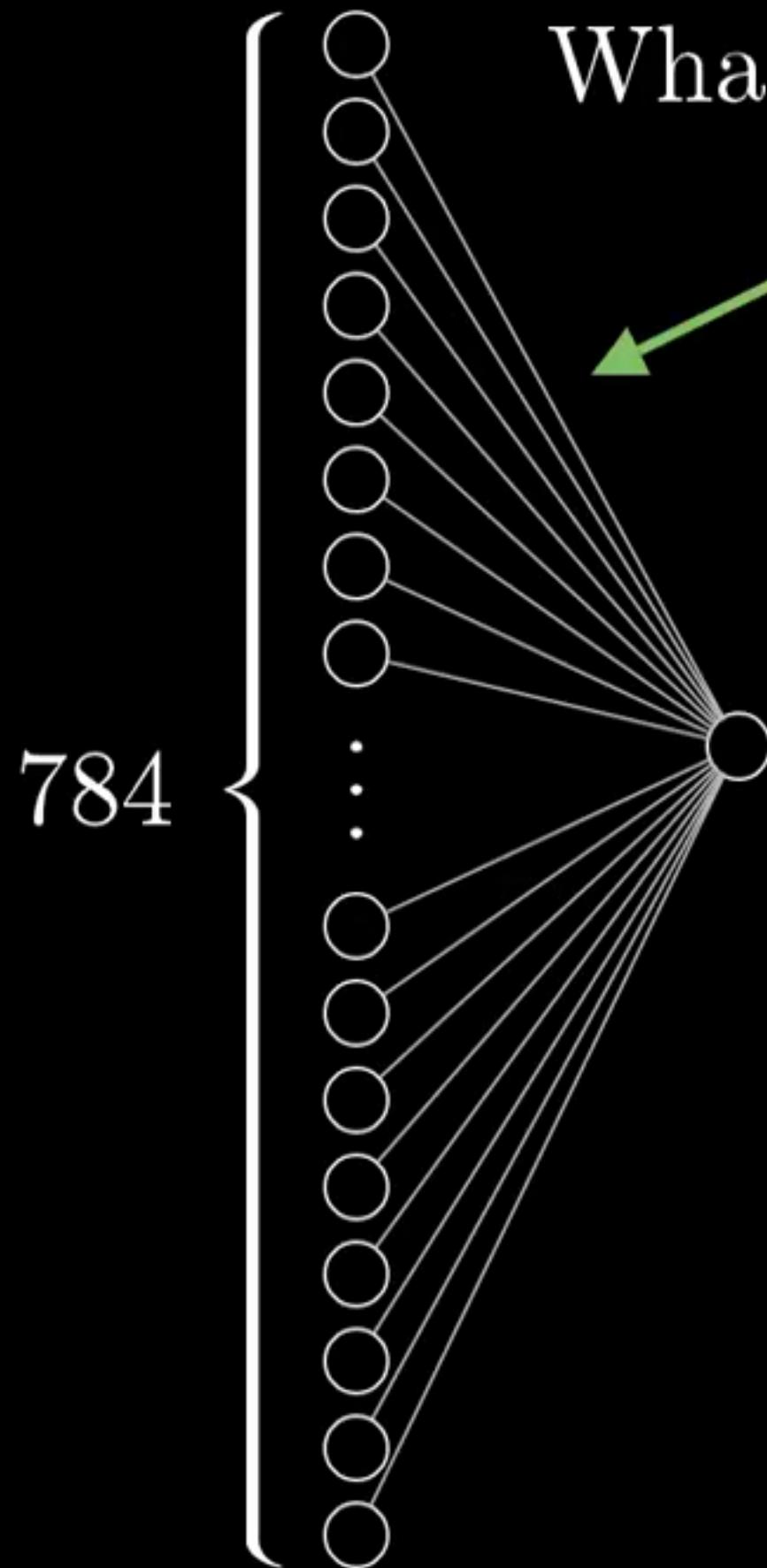
What parameters should exist?



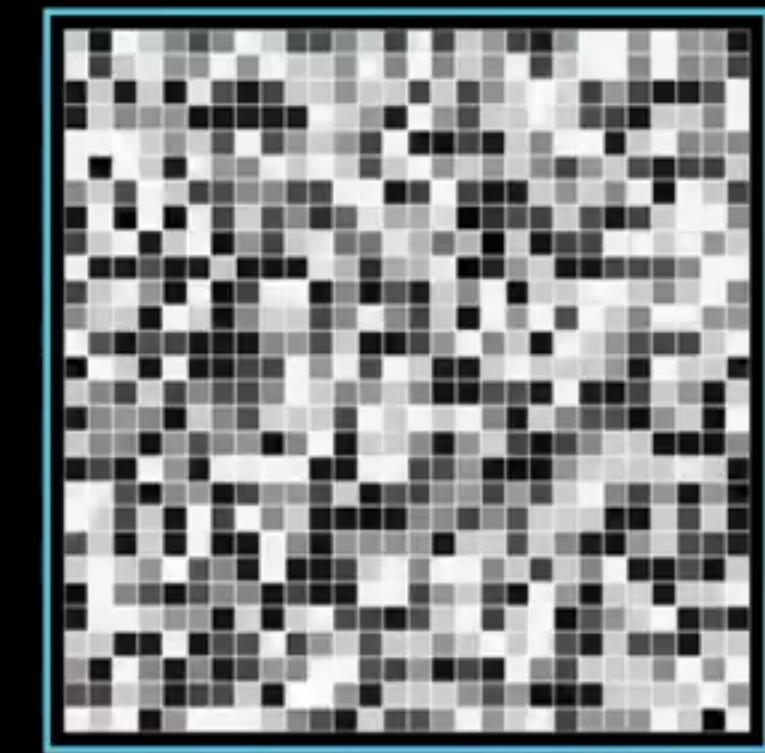
- $p_1: 0.88$
- $p_2: 0.63$
- $p_3: 0.98$
- $p_4: 1.00$
- $p_5: 0.17$
- $p_6: 0.71$
- $p_7: 0.98$
- $p_8: 0.43$
- \vdots



What parameters should exist?

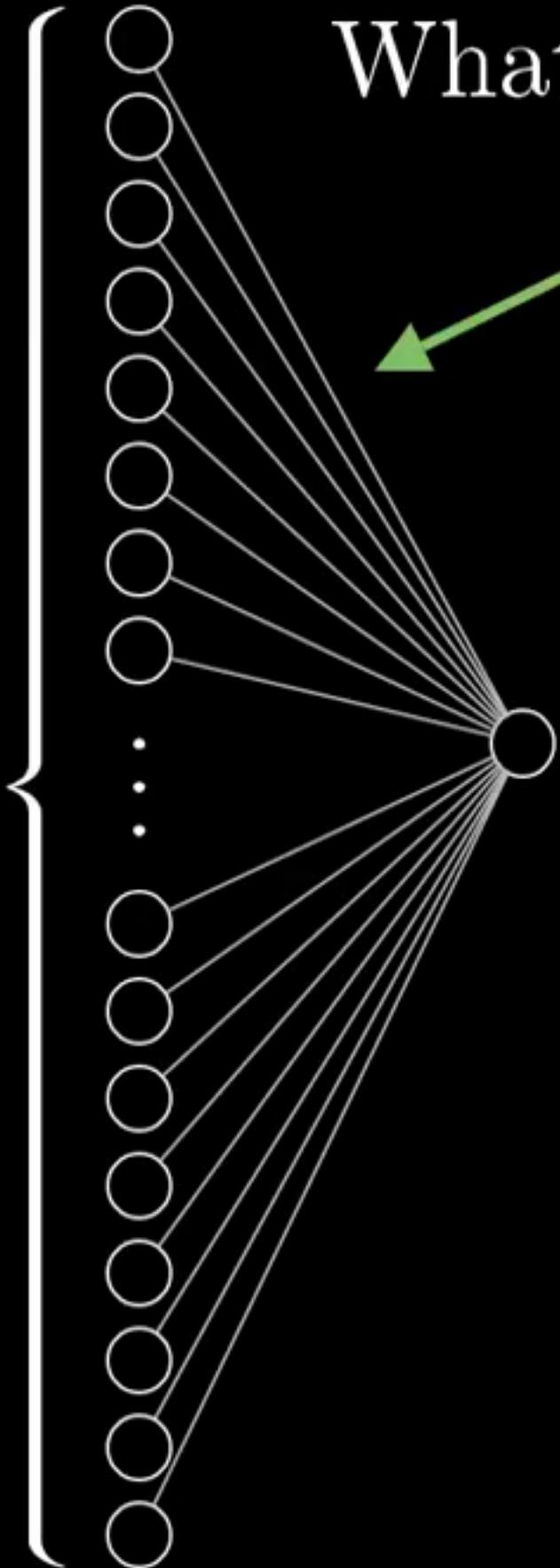


- $p_1: 0.98$
- $p_2: 0.27$
- $p_3: 0.31$
- $p_4: 0.06$
- $p_5: 0.99$
- $p_6: 0.59$
- $p_7: 0.31$
- $p_8: 0.08$
- \vdots

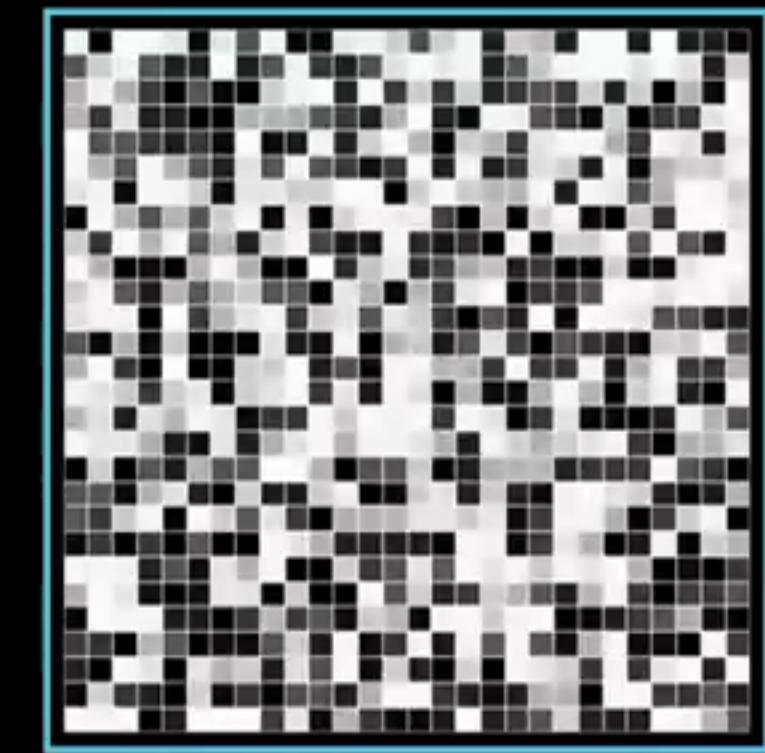


What parameters should exist?

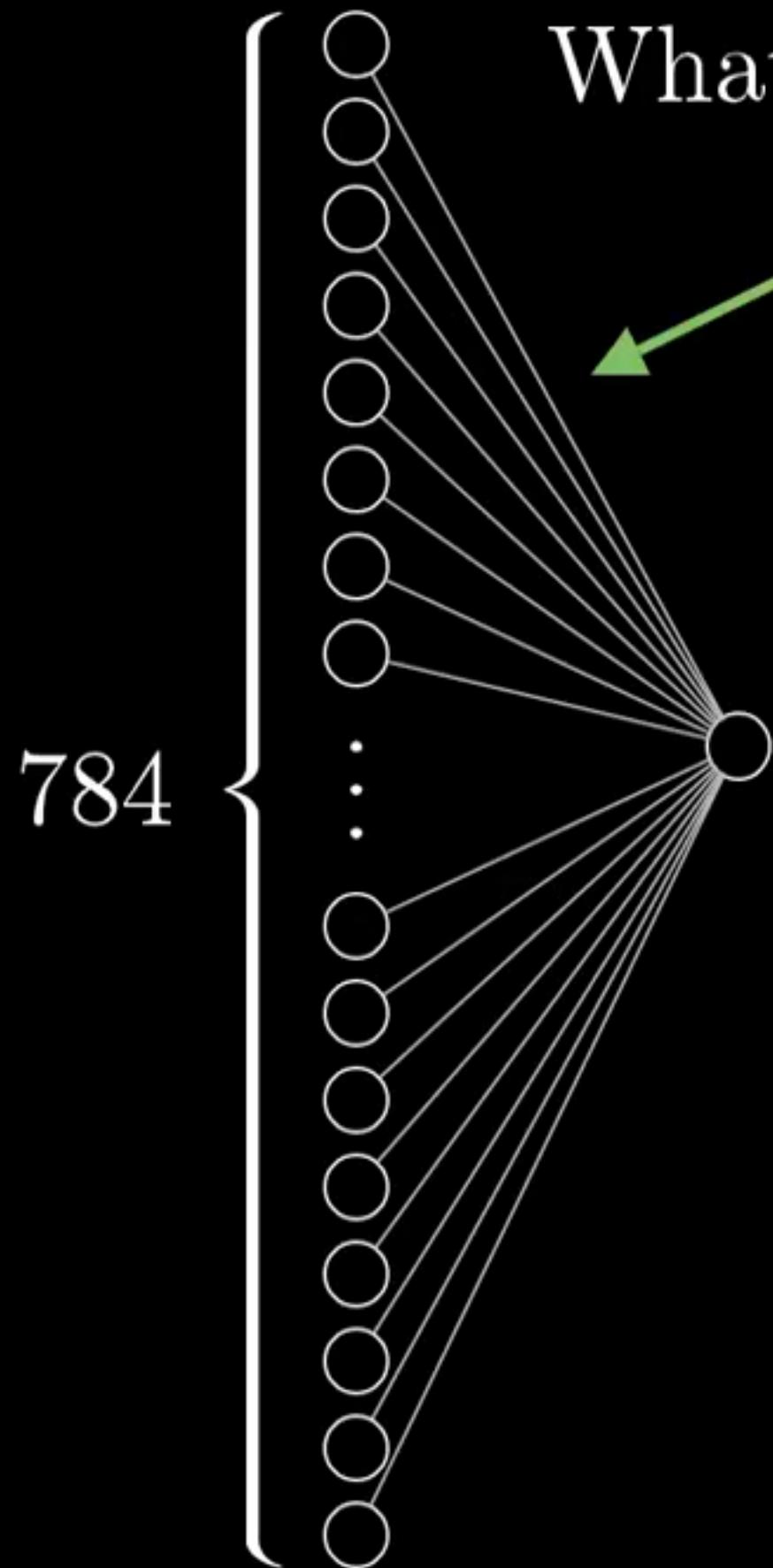
784



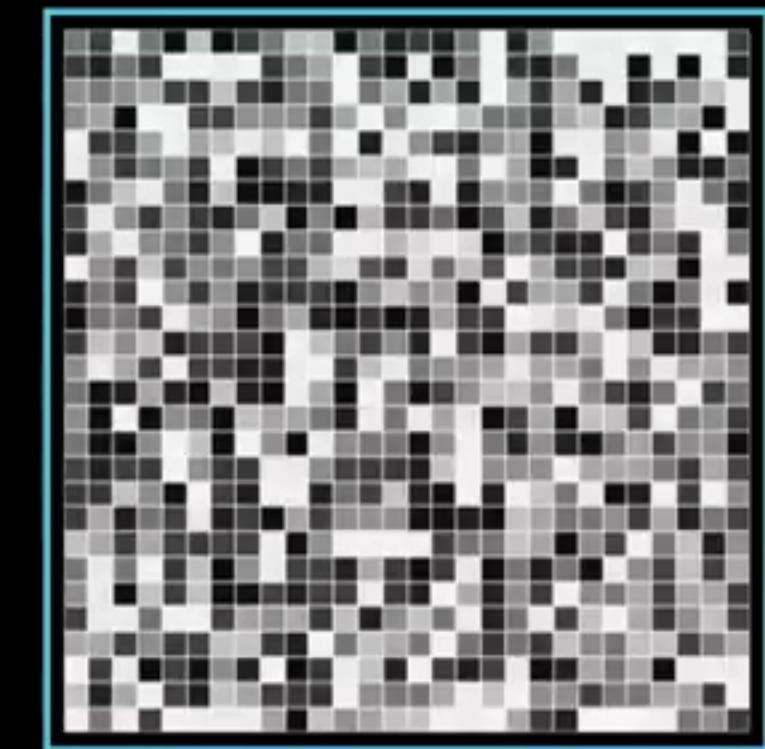
- $p_1: 1.00$
- $p_2: 0.82$
- $p_3: 0.06$
- $p_4: 0.71$
- $p_5: 0.35$
- $p_6: 0.14$
- $p_7: 0.06$
- $p_8: 0.02$
- \vdots



What parameters should exist?

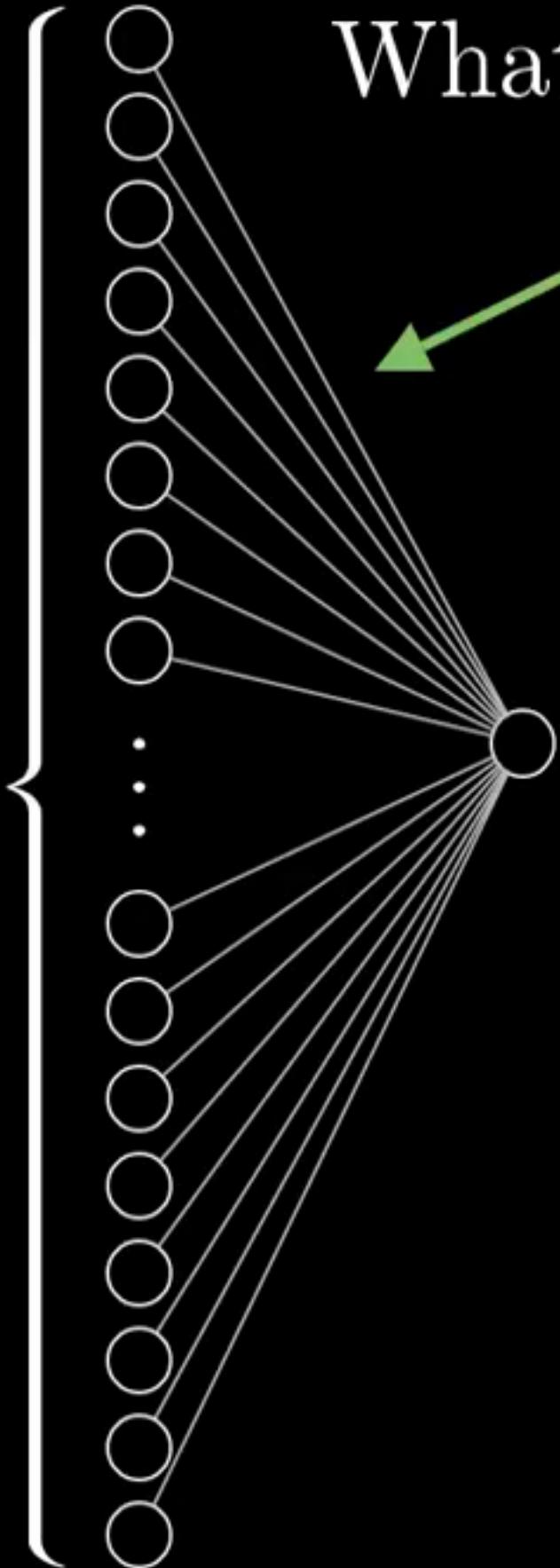


- $p_1: 0.93$
- $p_2: 0.11$
- $p_3: 0.79$
- $p_4: 0.59$
- $p_5: 0.17$
- $p_6: 1.00$
- $p_7: 0.79$
- $p_8: 0.27$
- \vdots

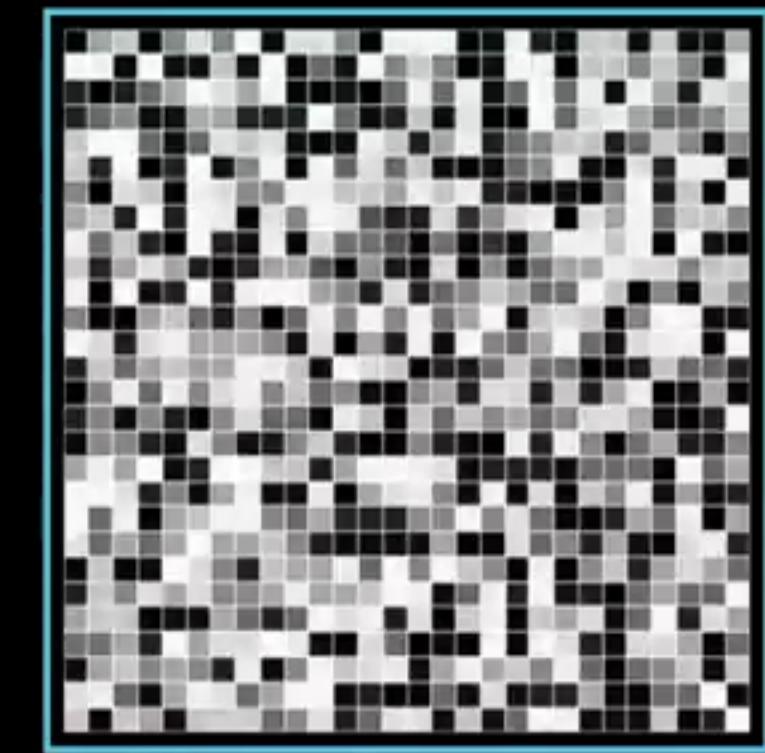


What parameters should exist?

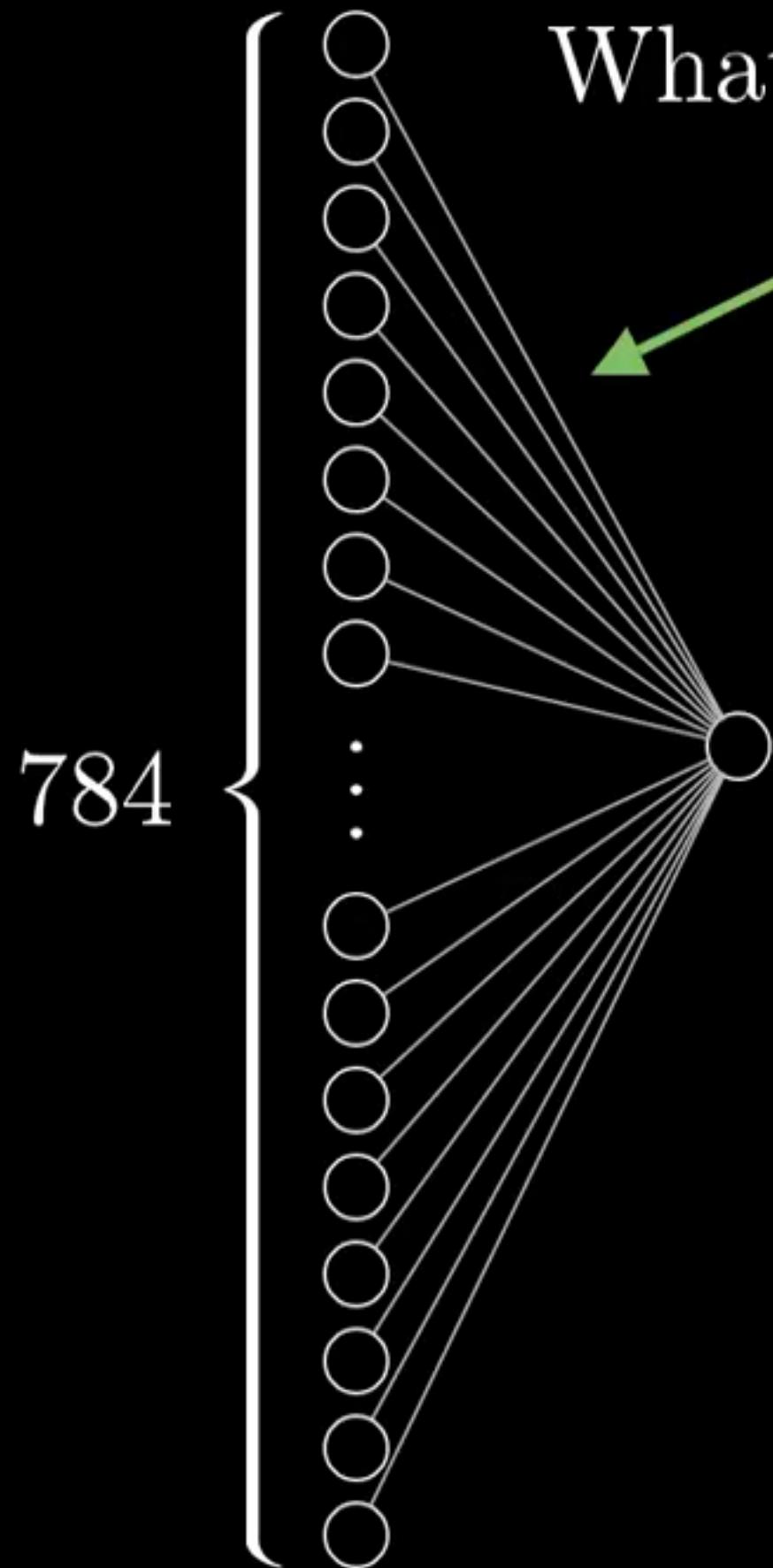
784



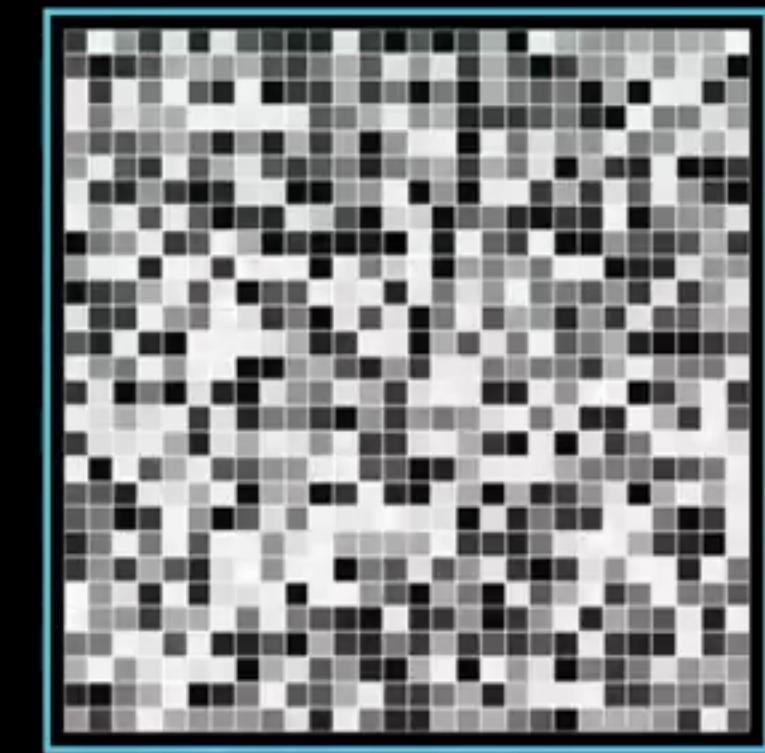
- $p_1: 0.79$
- $p_2: 0.95$
- $p_3: 0.88$
- $p_4: 0.14$
- $p_5: 0.99$
- $p_6: 0.06$
- $p_7: 0.88$
- $p_8: 0.67$
- \vdots



What parameters should exist?

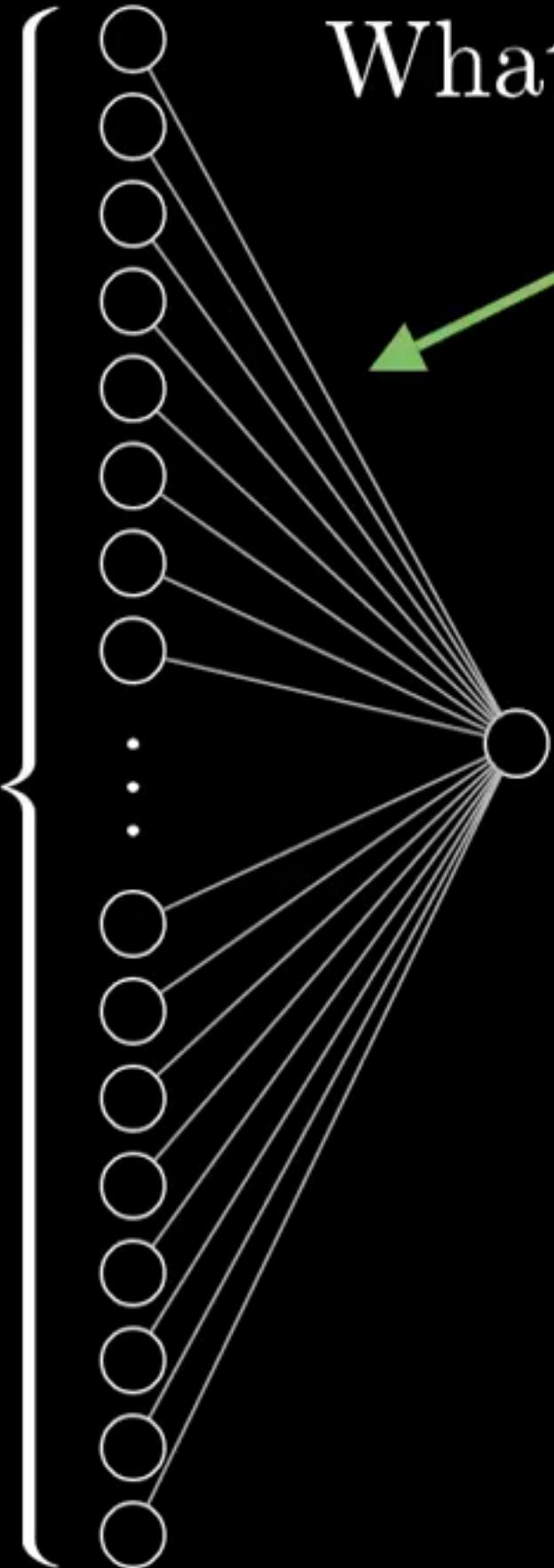


- $p_1: 0.59$
- $p_2: 0.02$
- $p_3: 0.14$
- $p_4: 1.00$
- $p_5: 0.35$
- $p_6: 0.71$
- $p_7: 0.14$
- $p_8: 0.96$
- \vdots

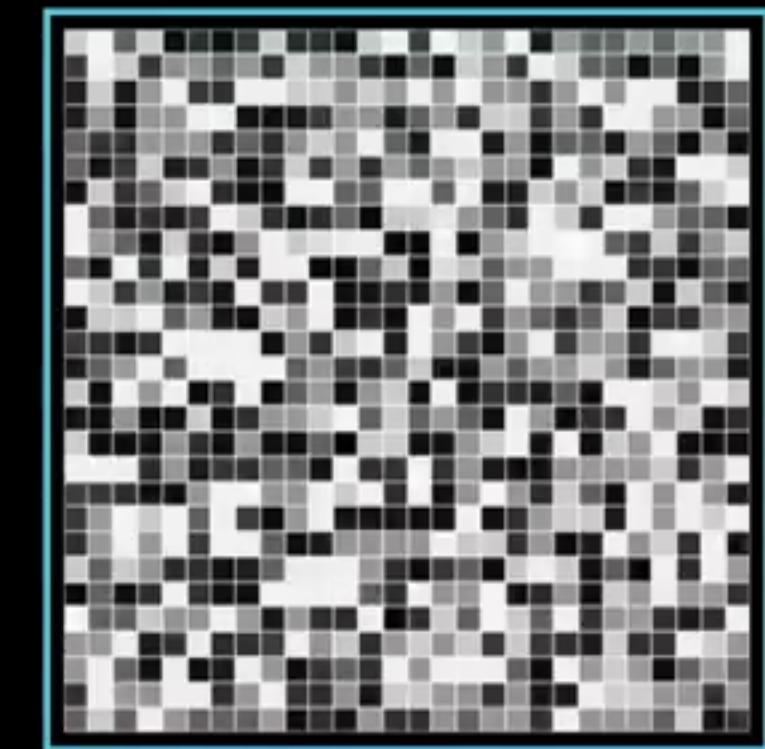


What parameters should exist?

784

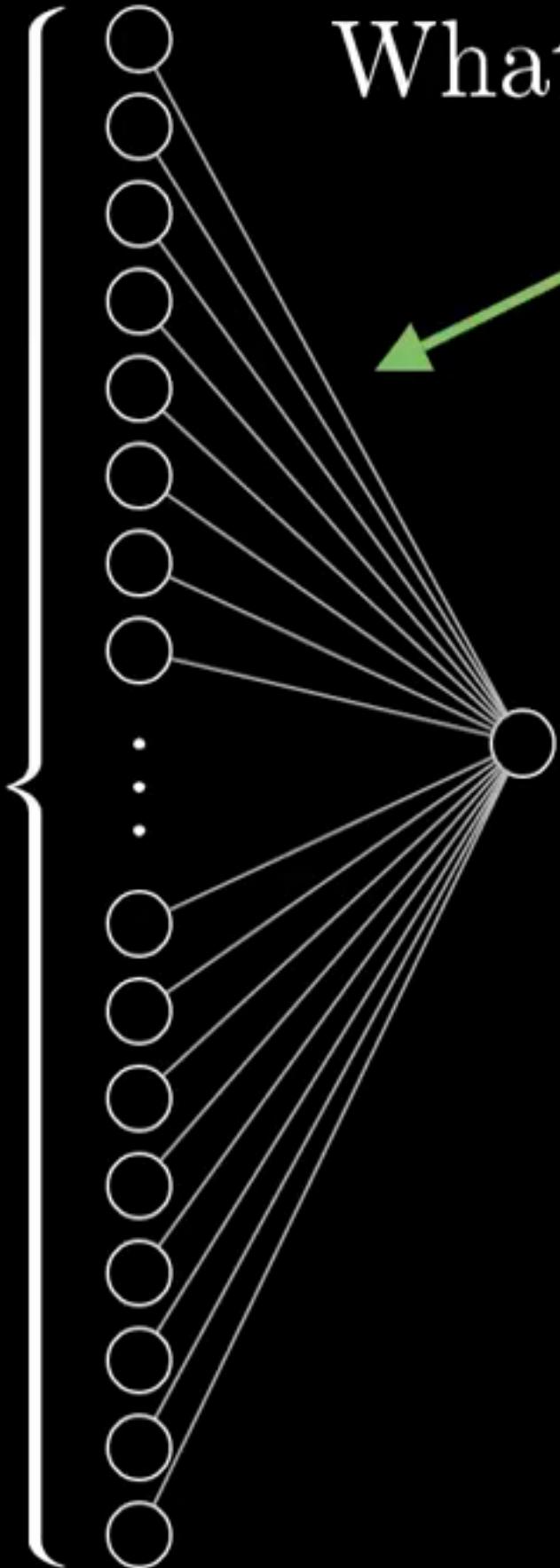


- $p_1: 0.39$
- $p_2: 1.00$
- $p_3: 0.20$
- $p_4: 0.06$
- $p_5: 0.17$
- $p_6: 0.59$
- $p_7: 0.20$
- $p_8: 0.95$
- \vdots

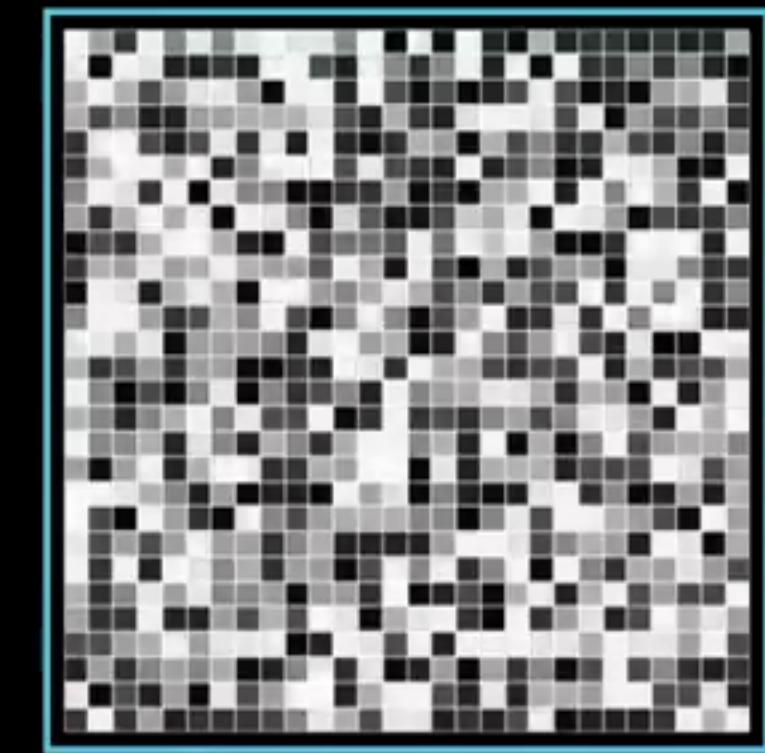


What parameters should exist?

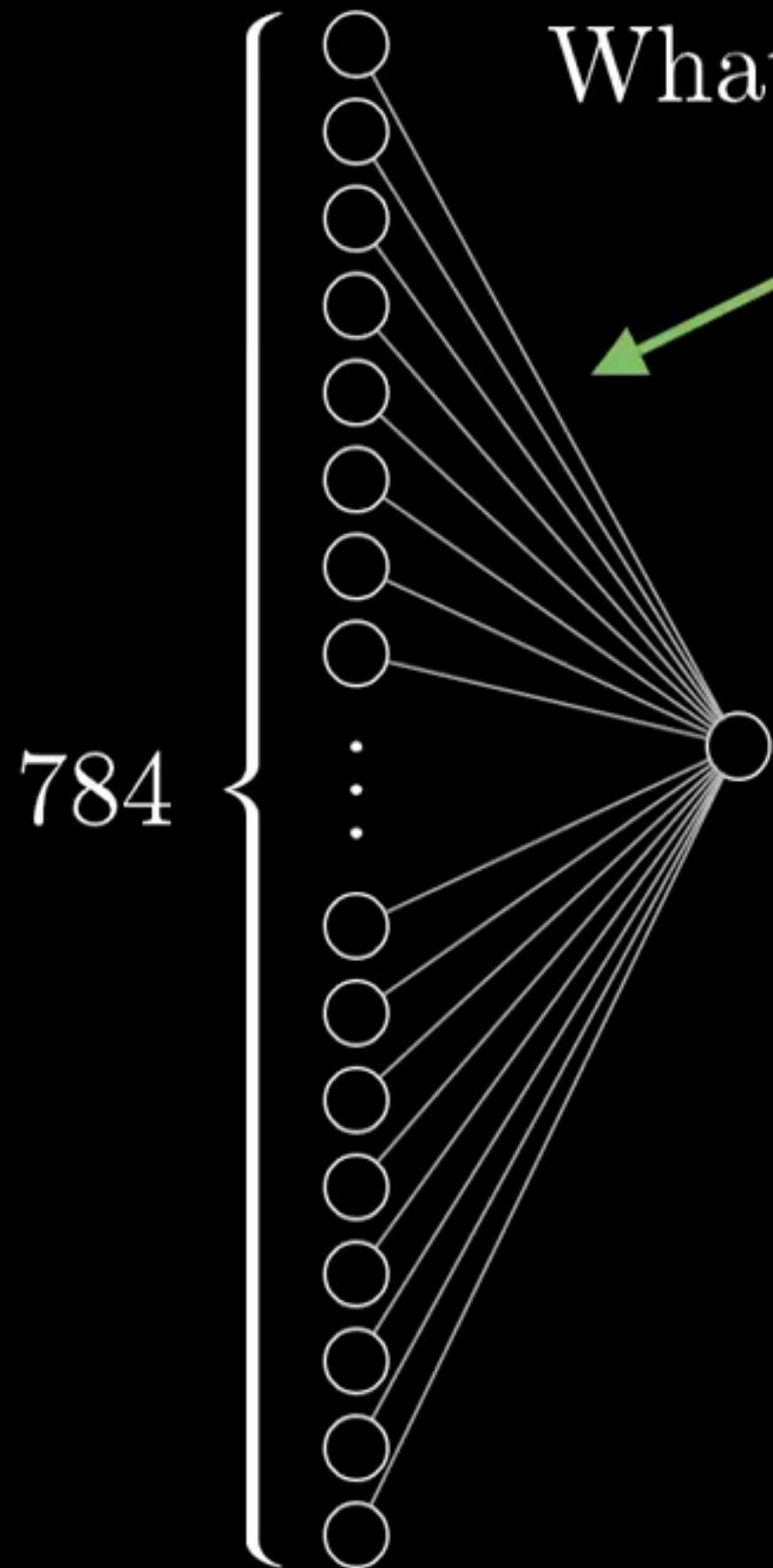
784



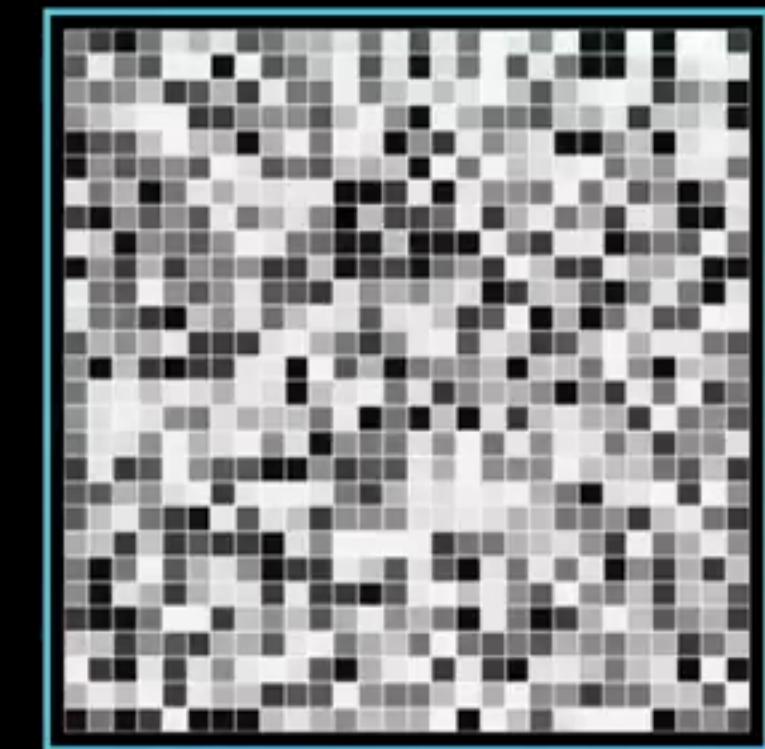
- $p_1: 0.20$
- $p_2: 0.01$
- $p_3: 0.93$
- $p_4: 0.71$
- $p_5: 0.99$
- $p_6: 0.14$
- $p_7: 0.93$
- $p_8: 0.63$
- \vdots



What parameters should exist?

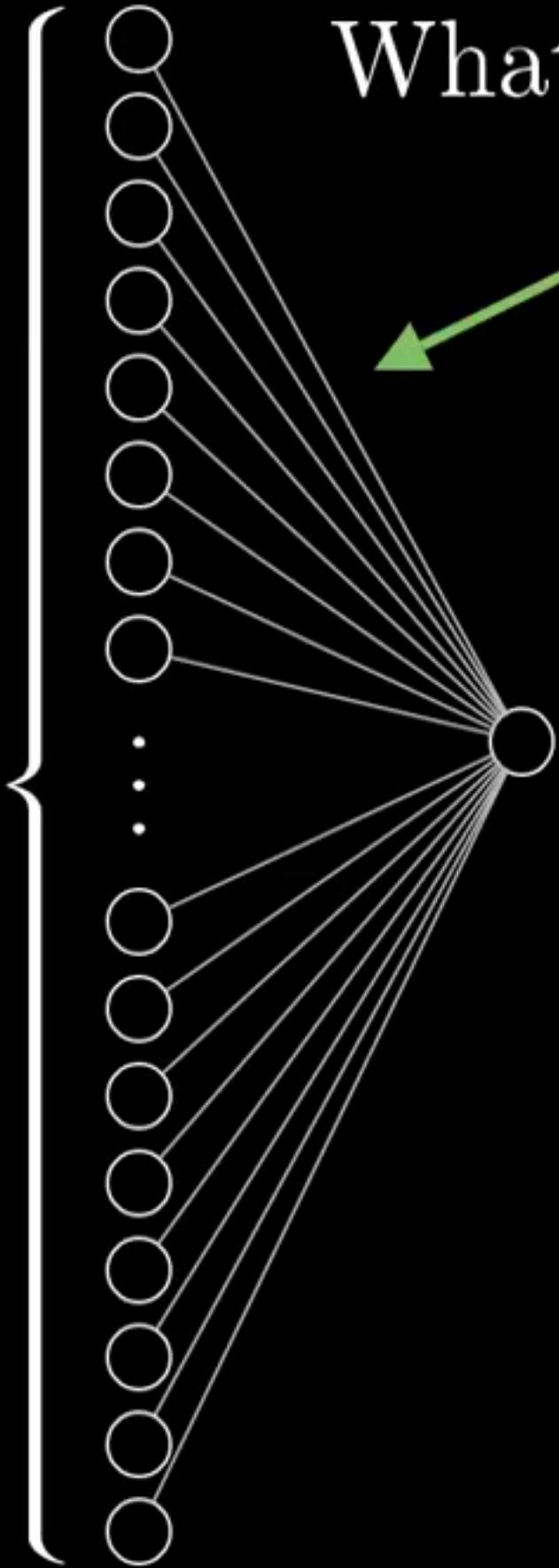


- $p_1: 0.06$
- $p_2: 0.96$
- $p_3: 0.71$
- $p_4: 0.59$
- $p_5: 0.35$
- $p_6: 1.00$
- $p_7: 0.71$
- $p_8: 0.23$
- \vdots

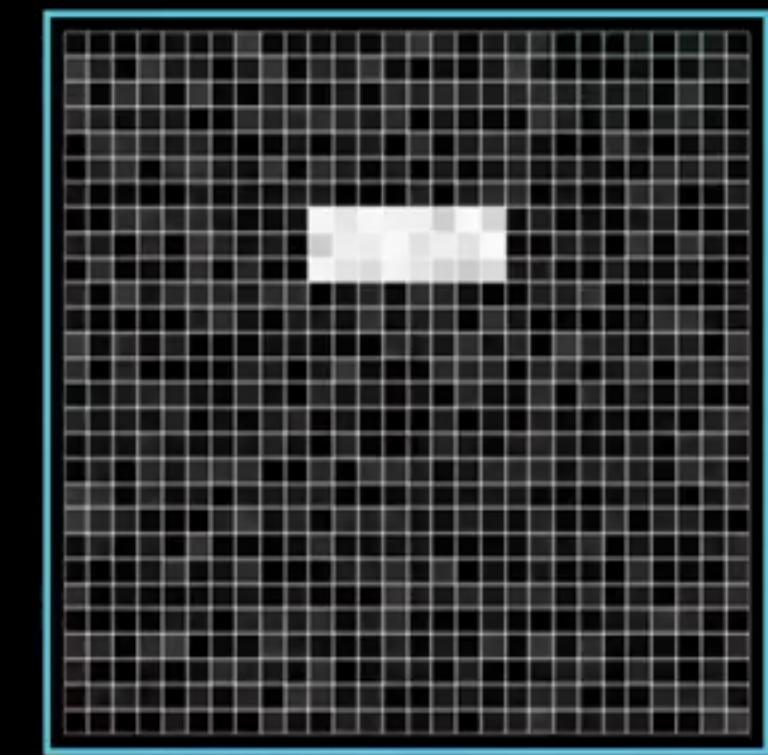


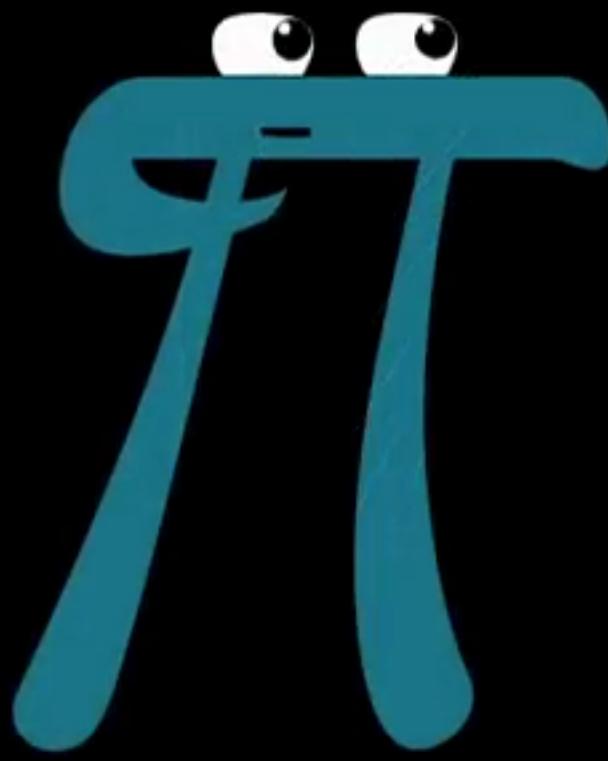
What parameters should exist?

784



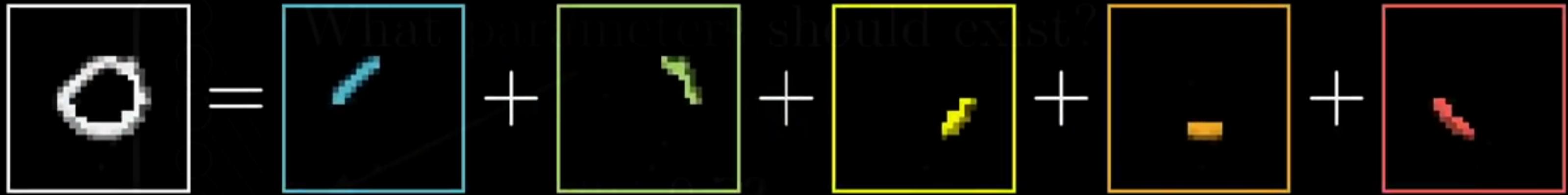
- $p_1: 0.00$
- $p_2: 0.08$
- $p_3: 0.03$
- $p_4: 0.14$
- $p_5: 0.17$
- $p_6: 0.06$
- $p_7: 0.03$
- $p_8: 0.01$
- \vdots



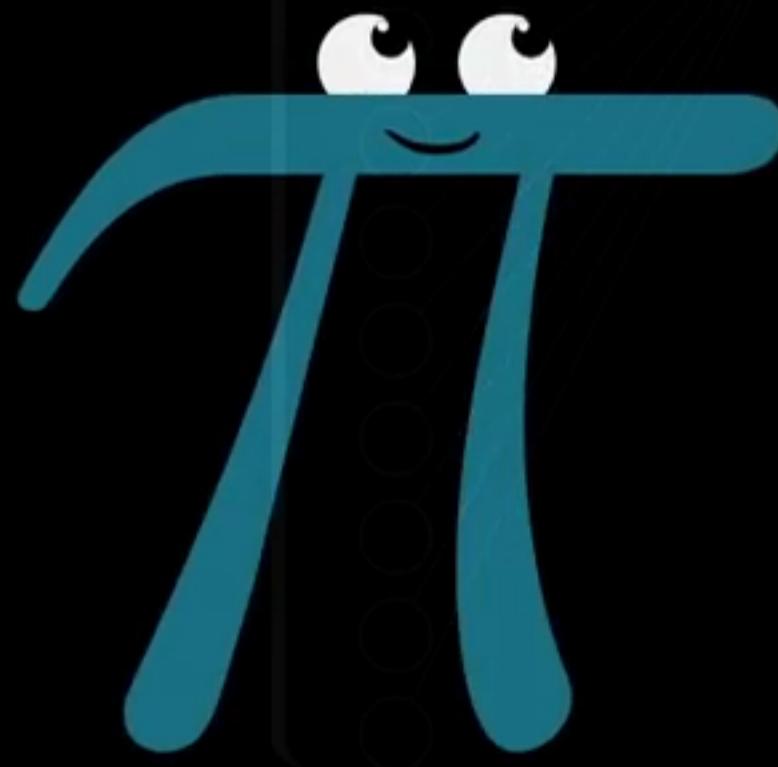


—



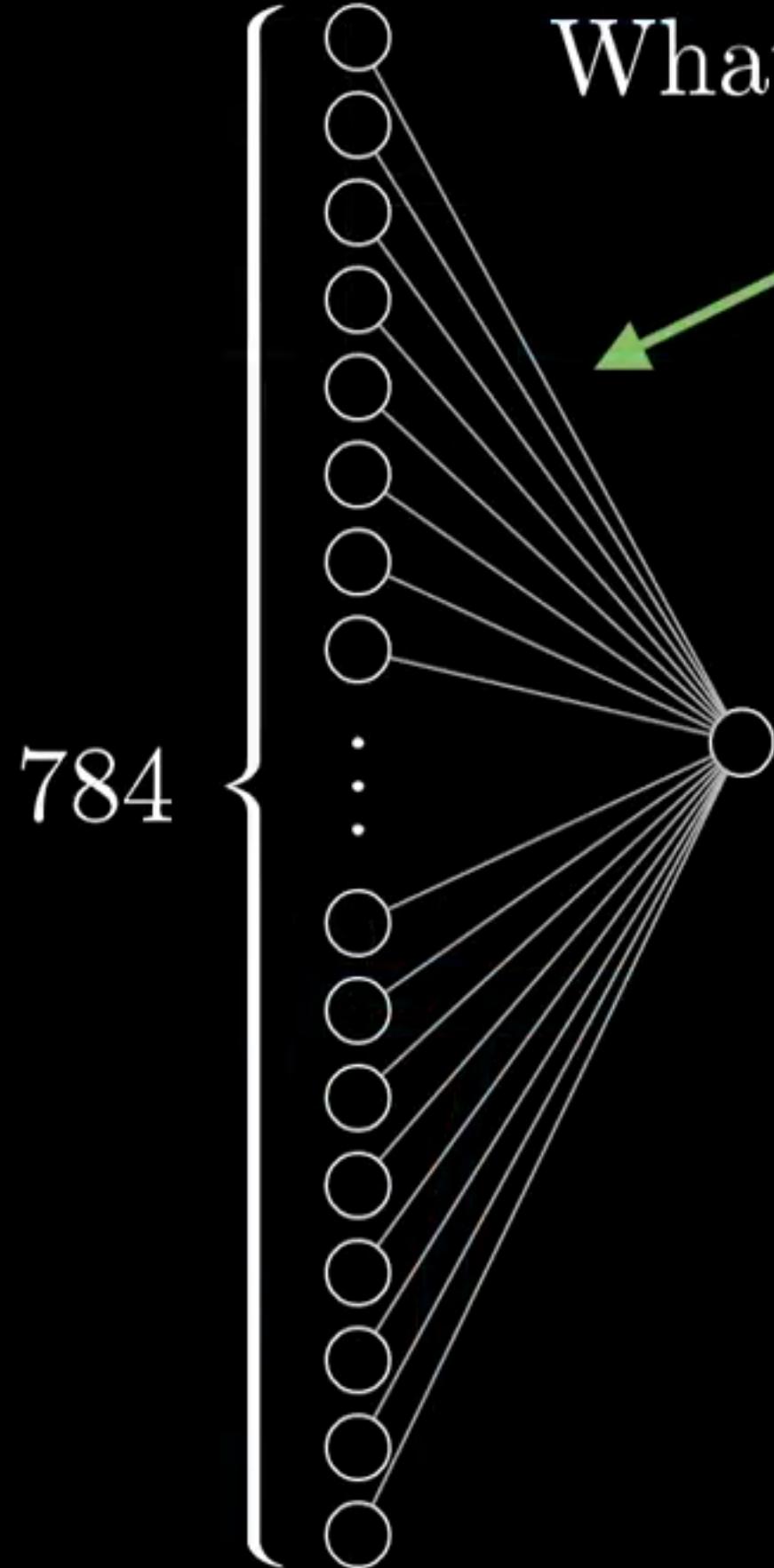


784



$P_1 = 0.73$
 $P_2 = 0.73$
 $P_3 = 0.73$
 $P_4 = 0.73$
 $P_5 = 0.79$
 $P_6 = 0.79$
 $P_7 = 0.79$
 $P_8 = 0.79$

What parameters should exist?

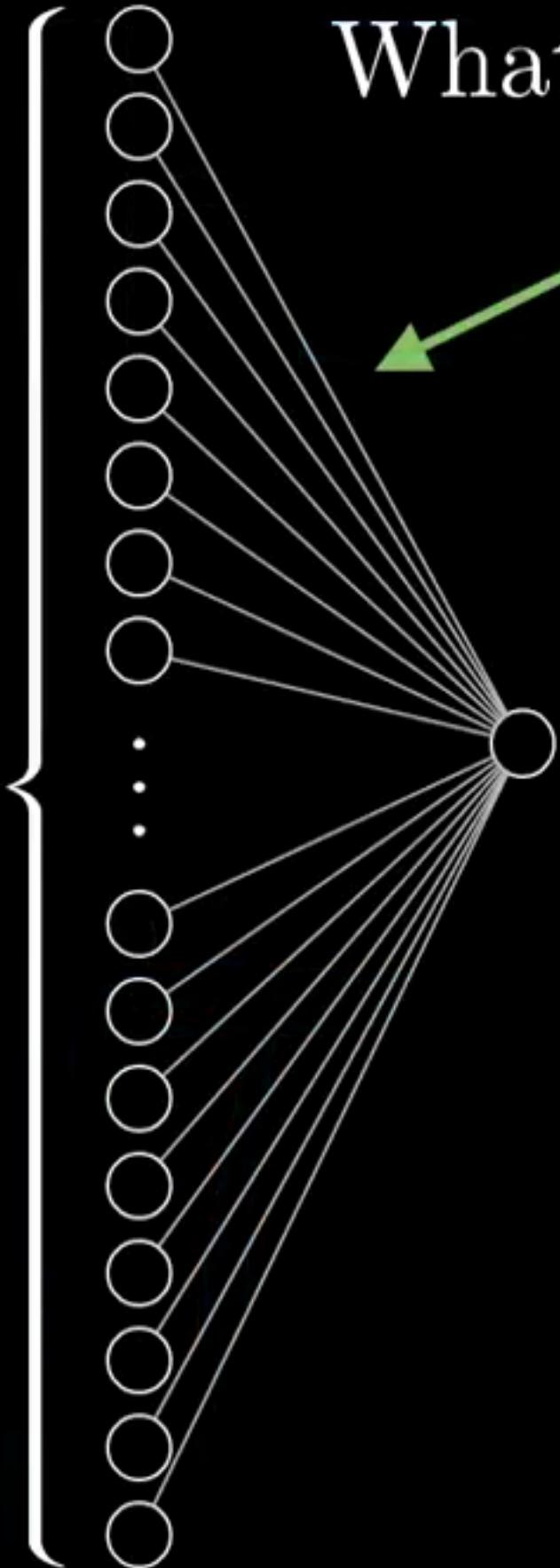


- $p_1: 0.15$
- $p_2: 0.15$
- $p_3: 0.15$
- $p_4: 0.15$
- $p_5: 0.51$
- $p_6: 0.51$
- $p_7: 0.51$
- $p_8: 0.51$
- \vdots

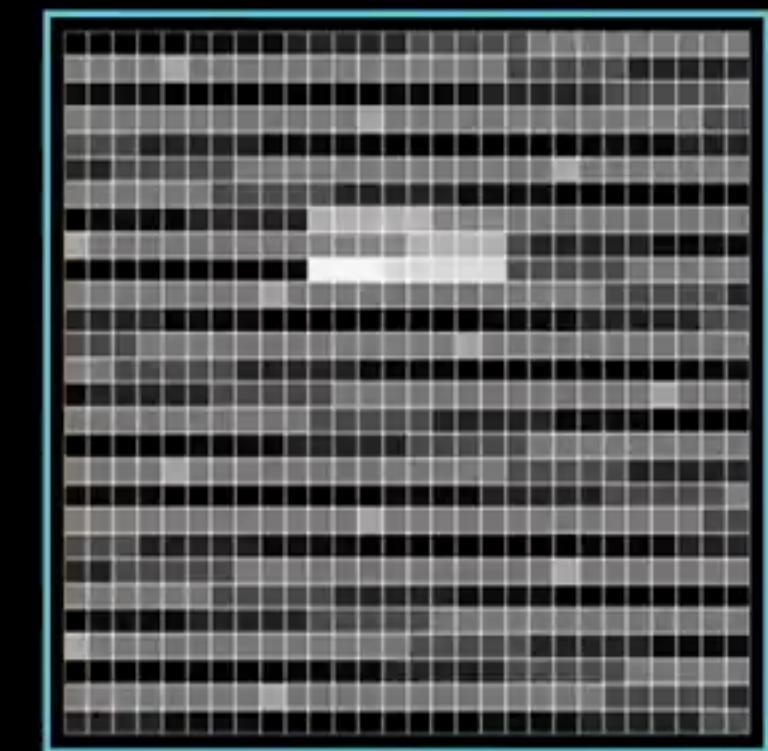


What parameters should exist?

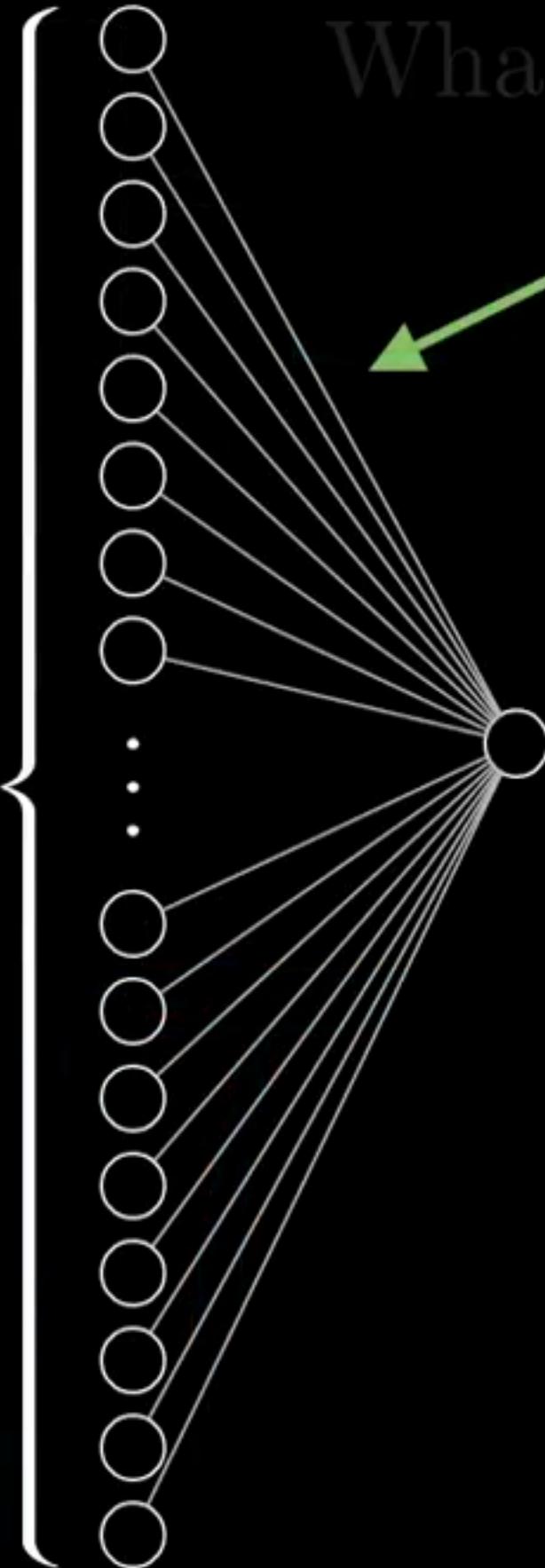
784



- $p_1: 0.04$
- $p_2: 0.04$
- $p_3: 0.04$
- $p_4: 0.04$
- $p_5: 0.14$
- $p_6: 0.14$
- $p_7: 0.14$
- $p_8: 0.14$
- \vdots



784



What **We**ights should exist?

$w_1: 0.00$

$w_2: 0.00$

*w*₃: 0.00

$w_4: 0.00$

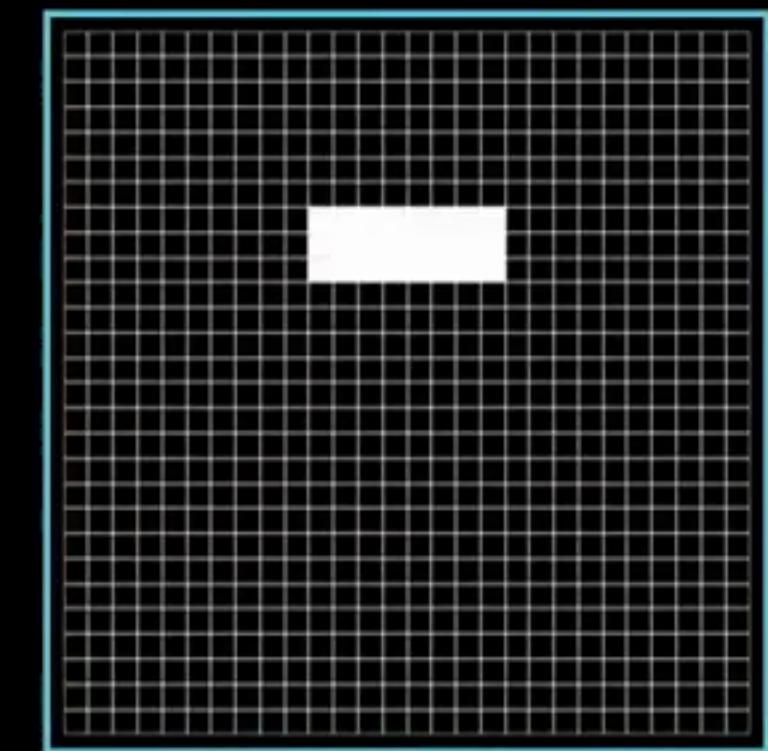
$w_5: 0.00$

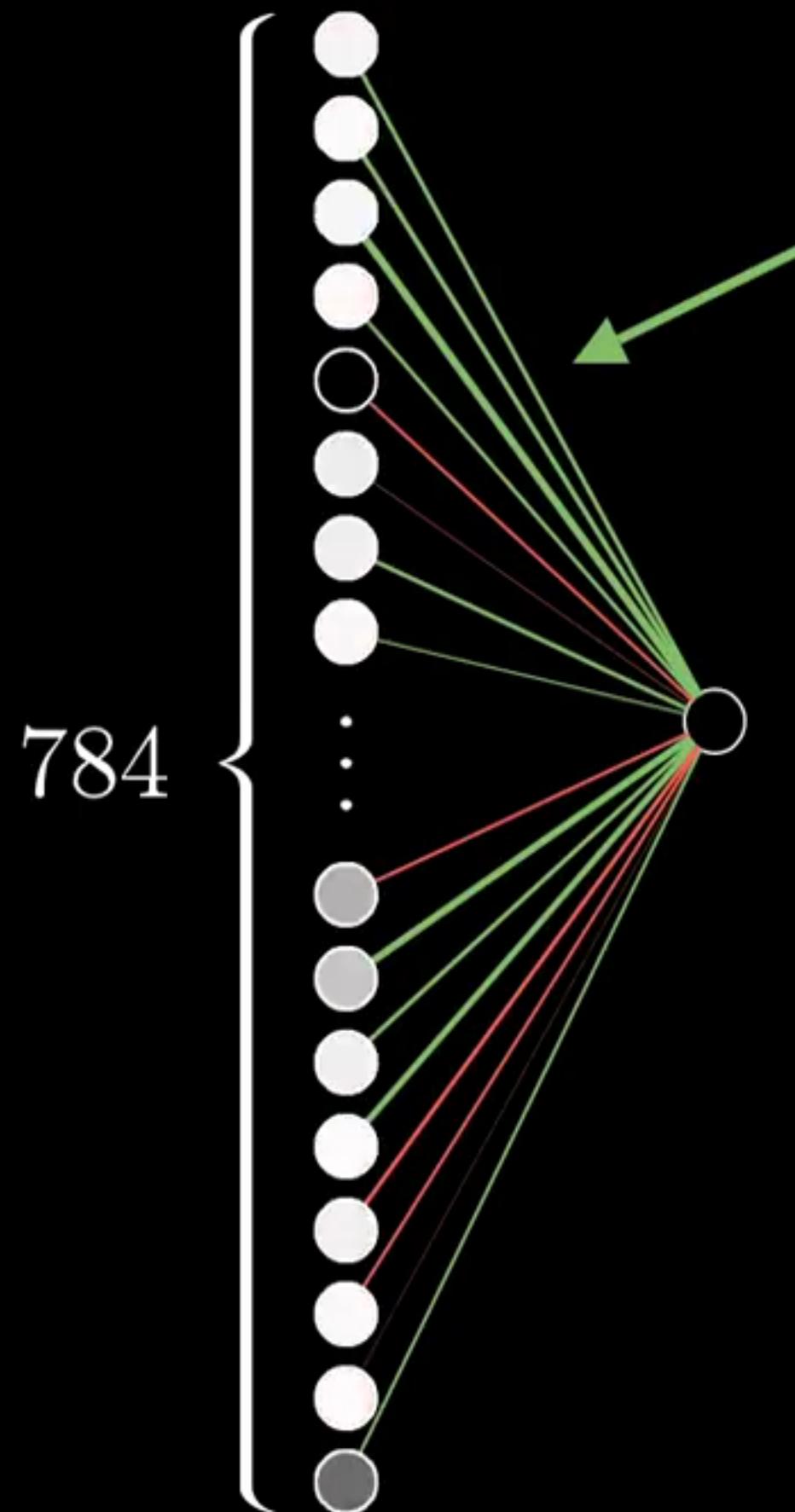
$w_6: 0.00$

$w_7: 0.00$

w8: 0.00

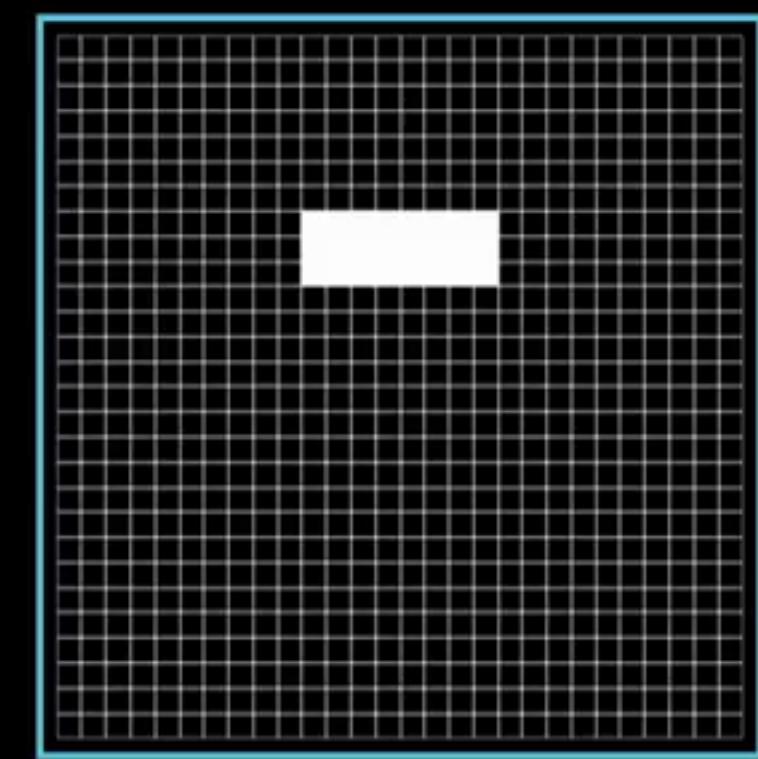
•

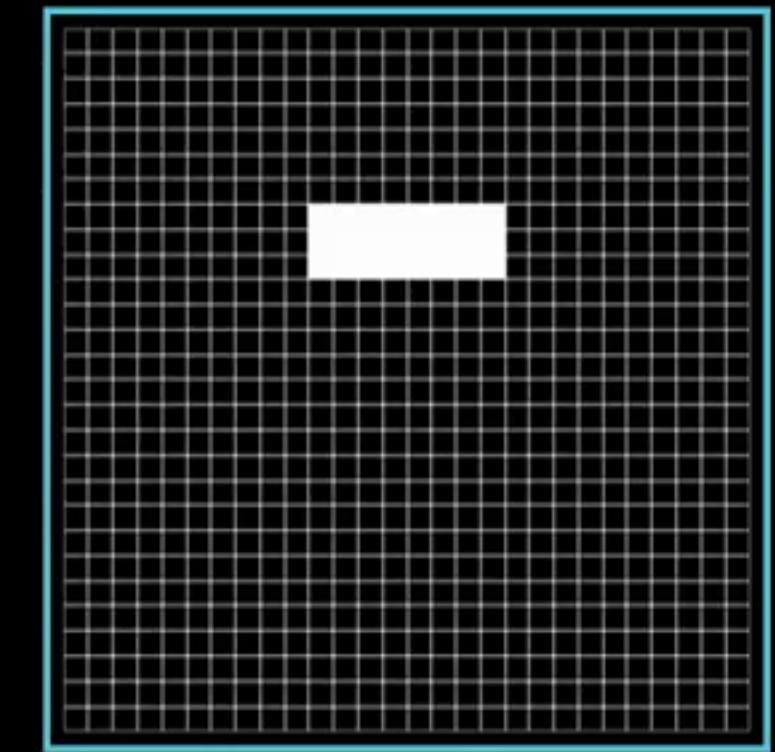
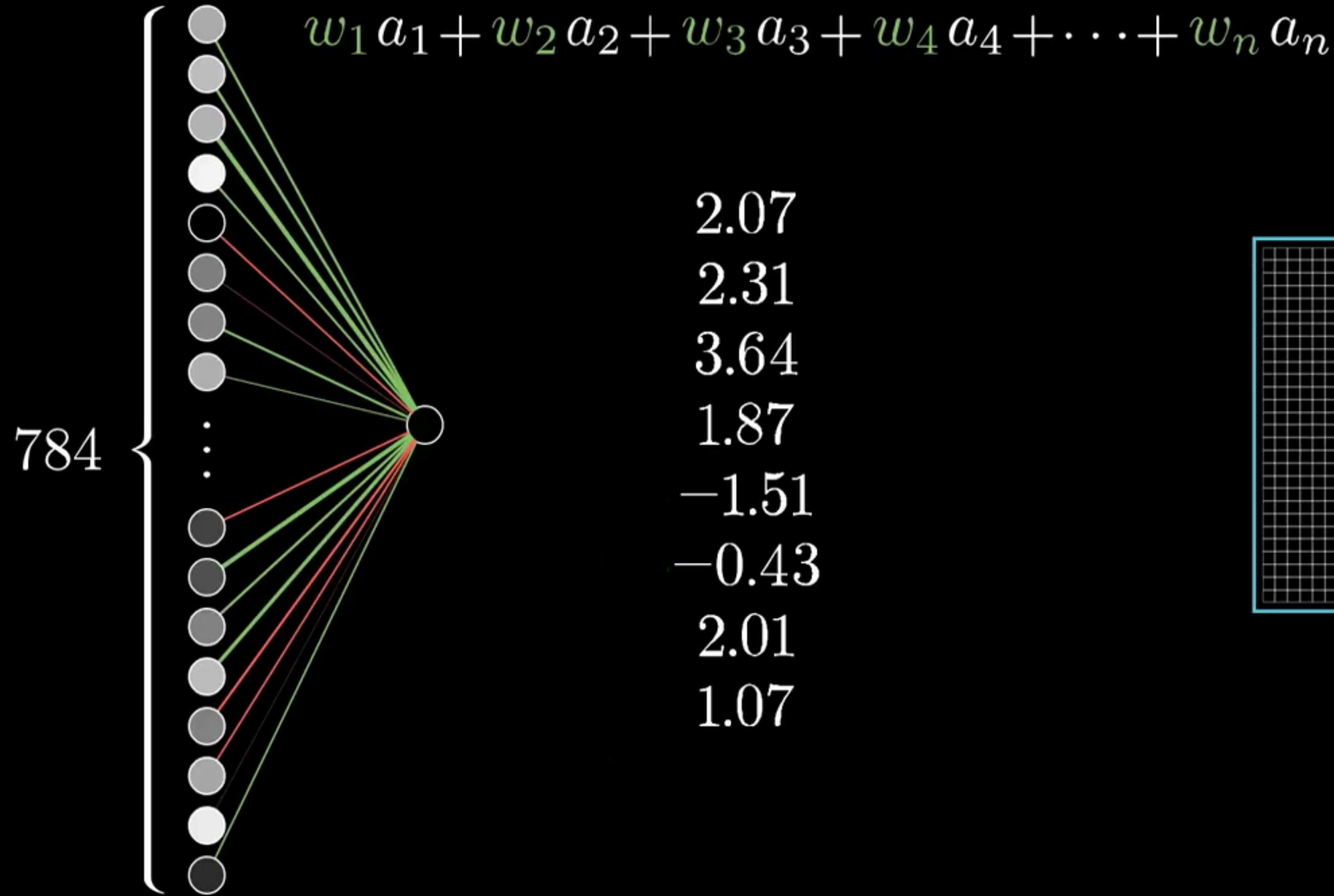


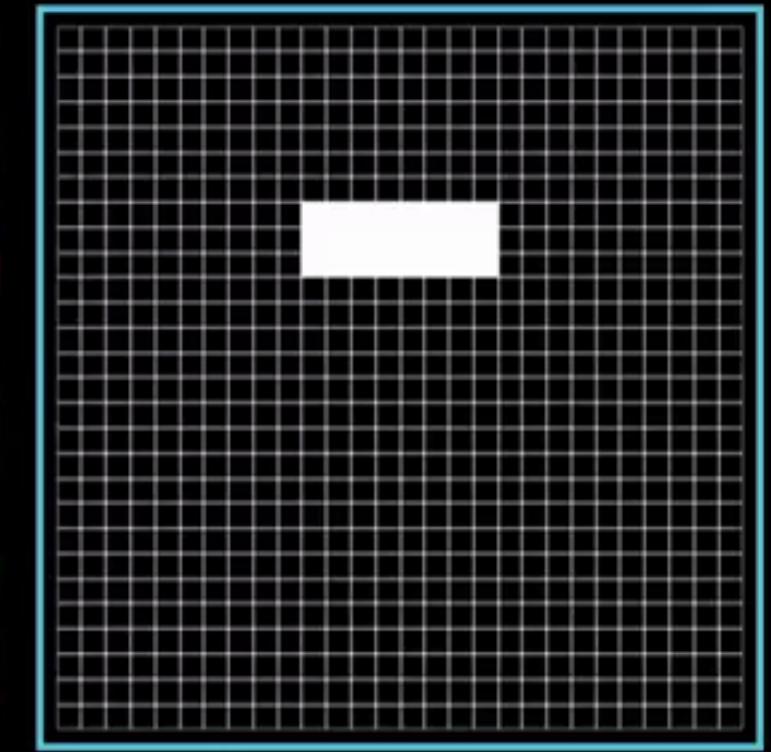
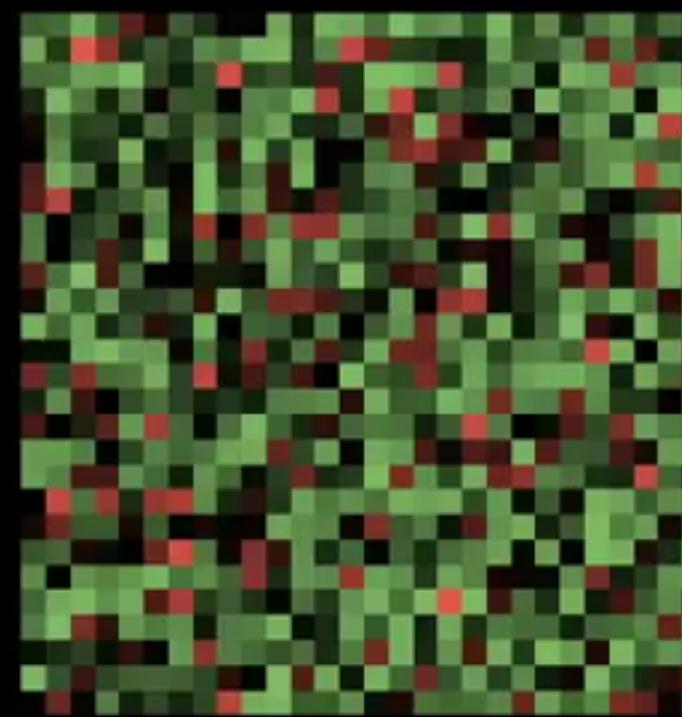
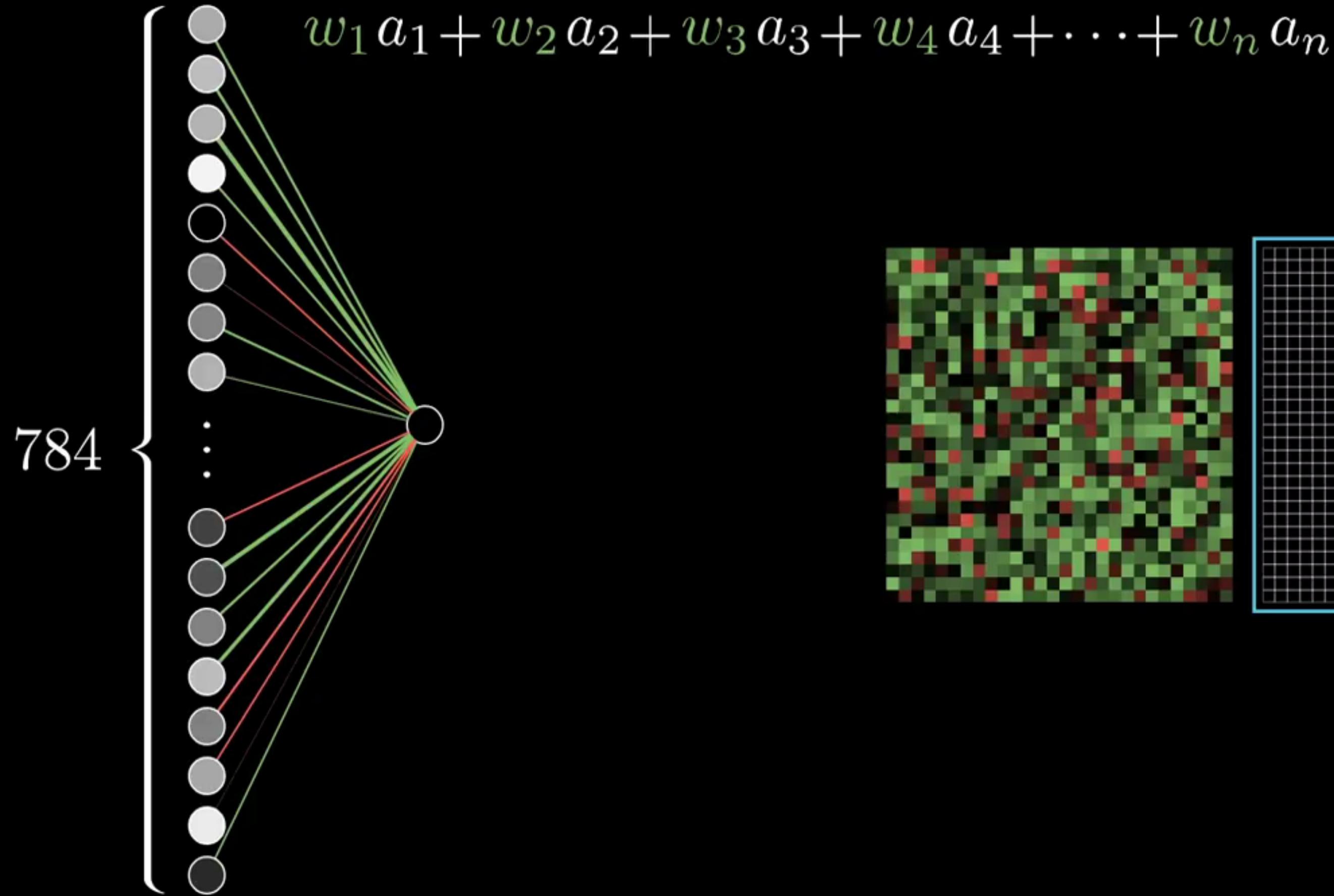


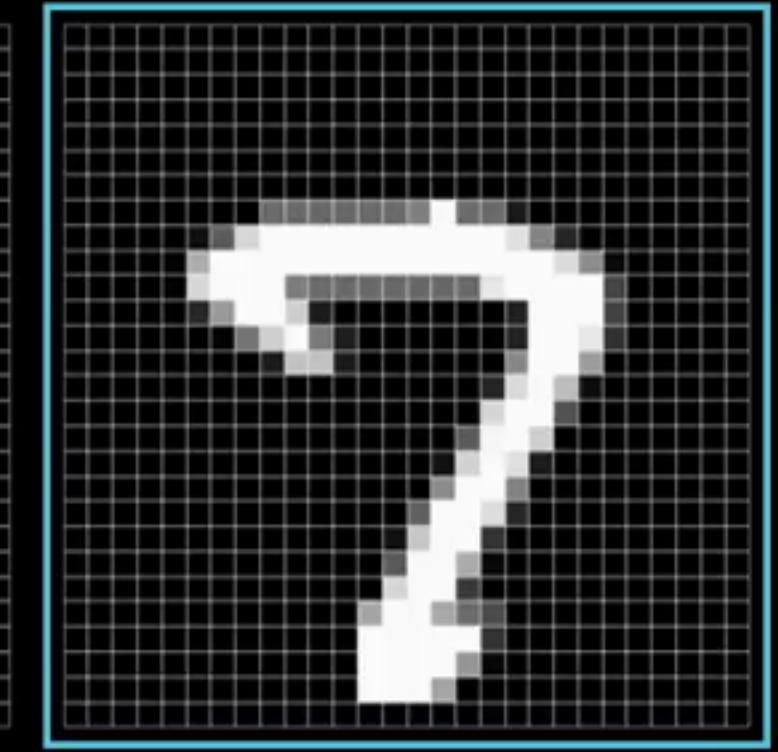
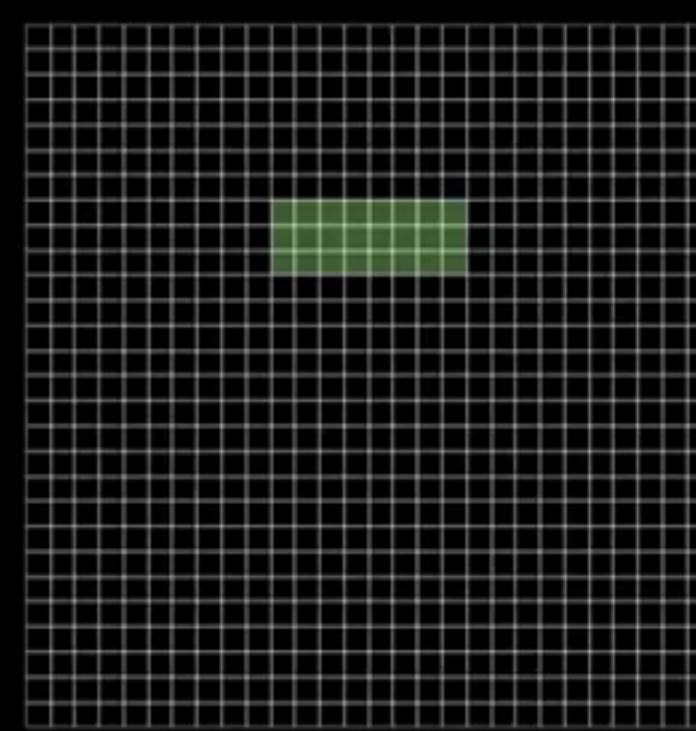
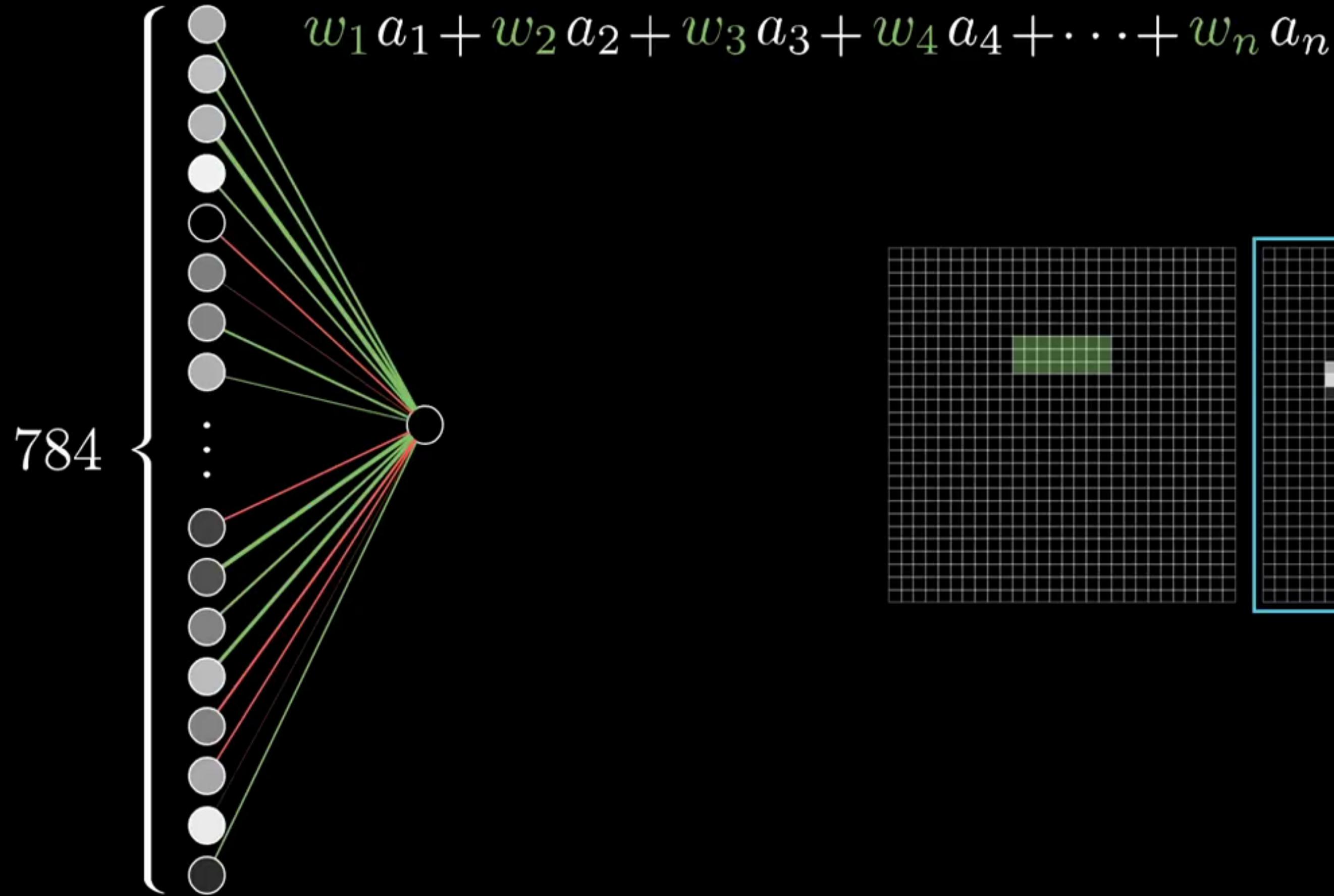
Weights

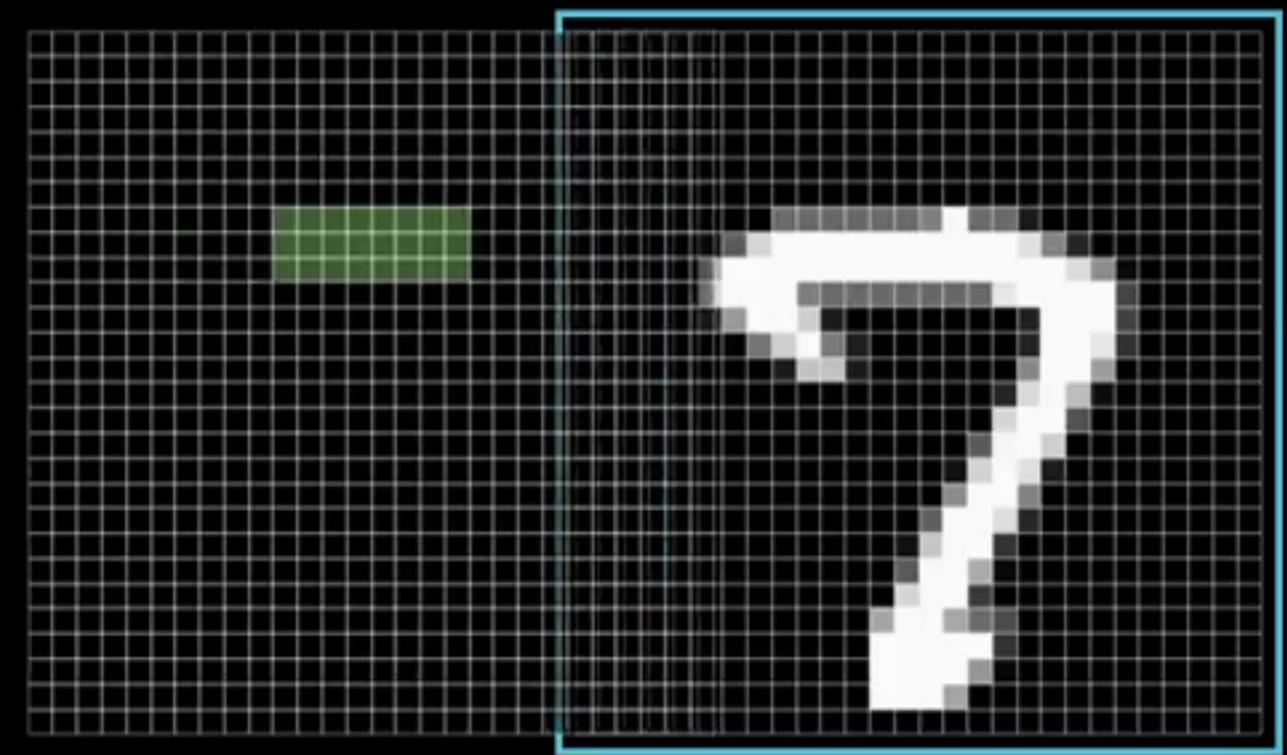
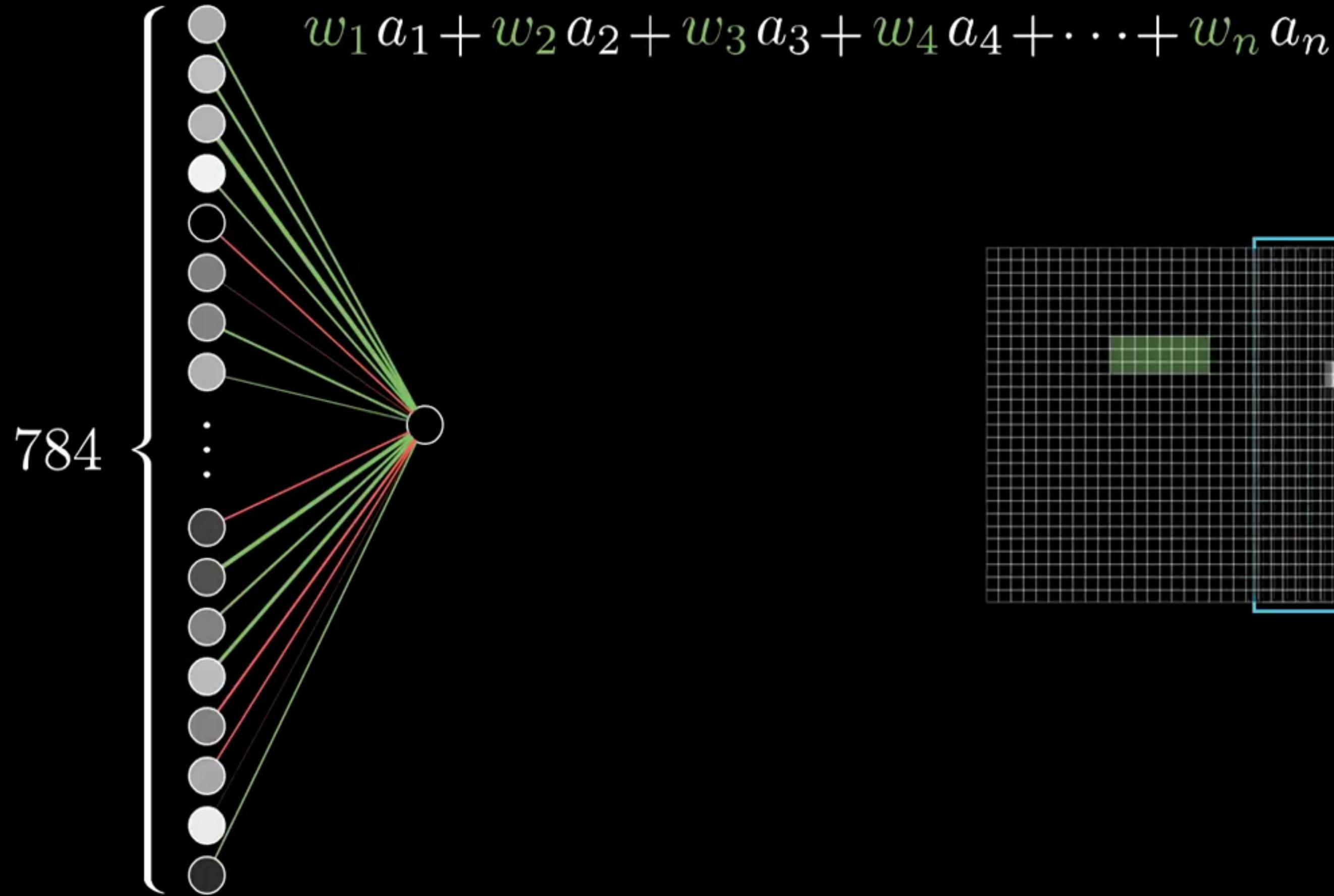
$w_1: 2.07$
 $w_2: 2.31$
 $w_3: 3.64$
 $w_4: 1.87$
 $w_5: -1.51$
 $w_6: -0.43$
 $w_7: 2.01$
 $w_8: 1.07$
⋮

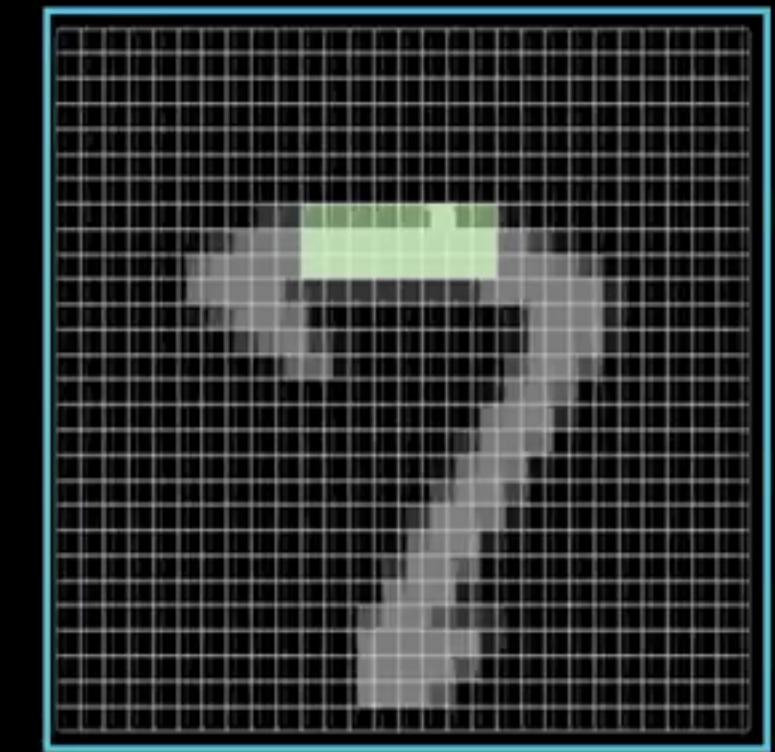
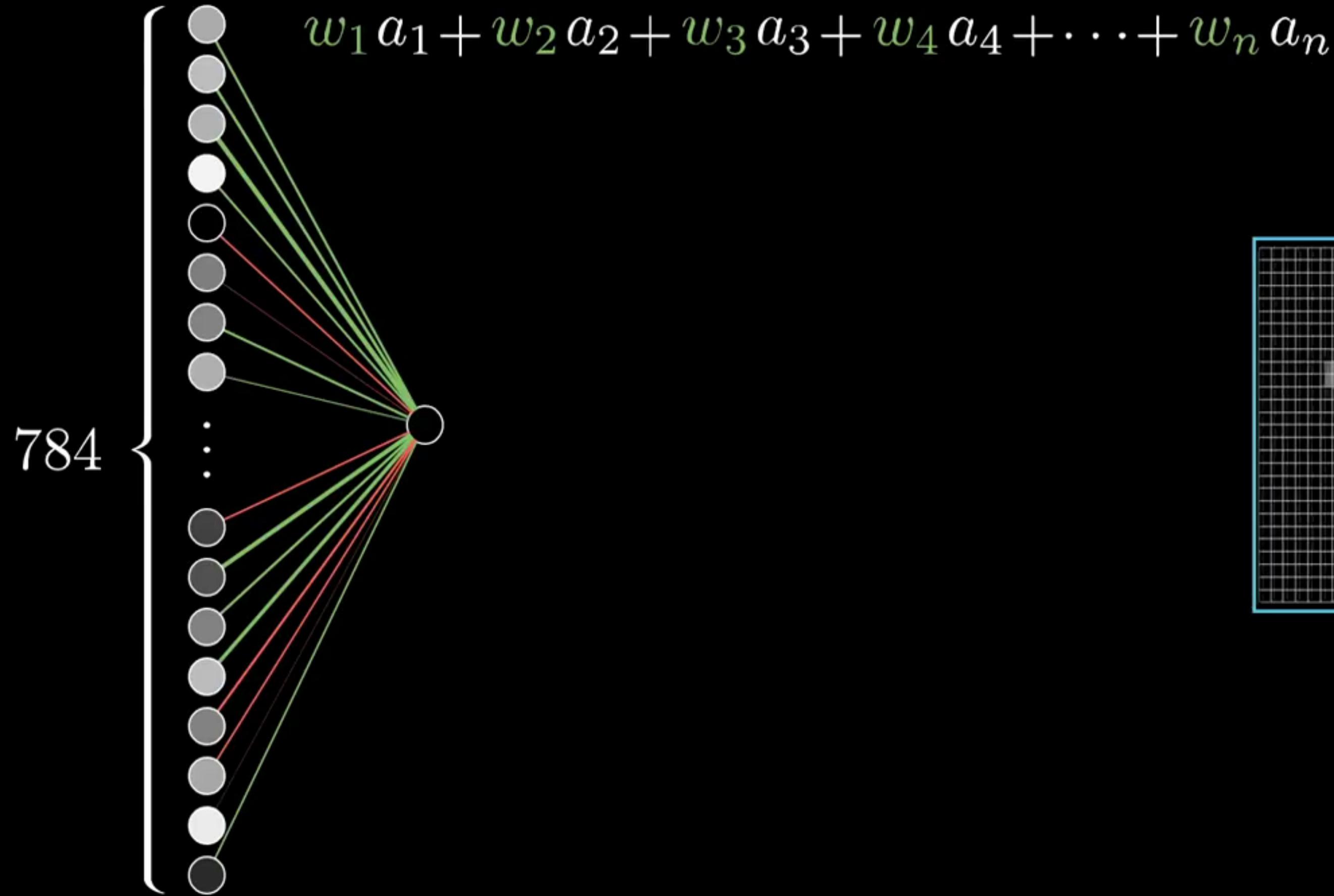


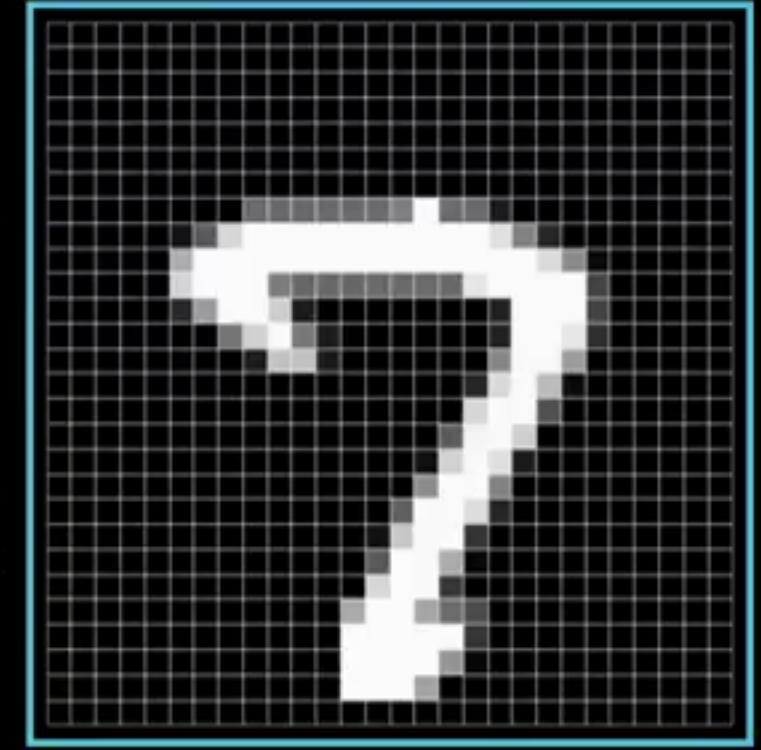
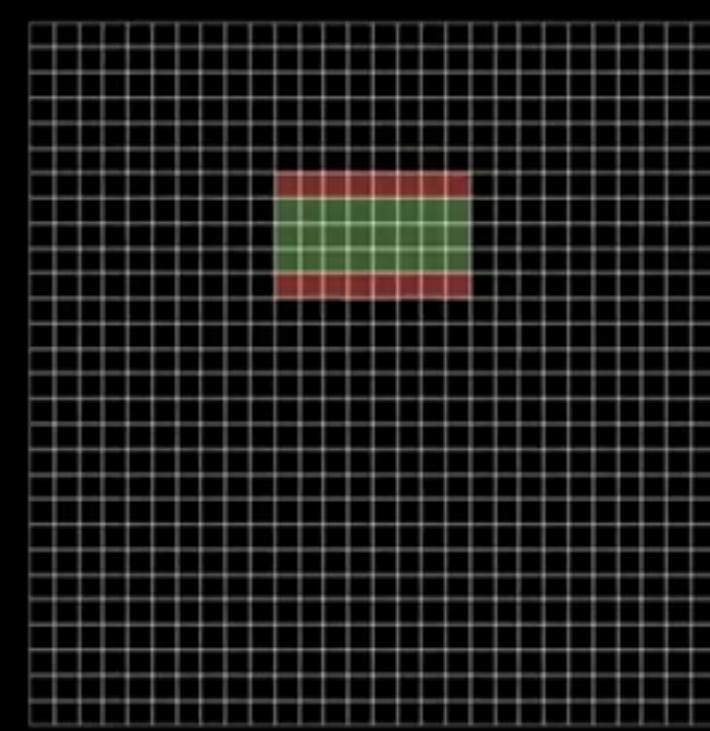
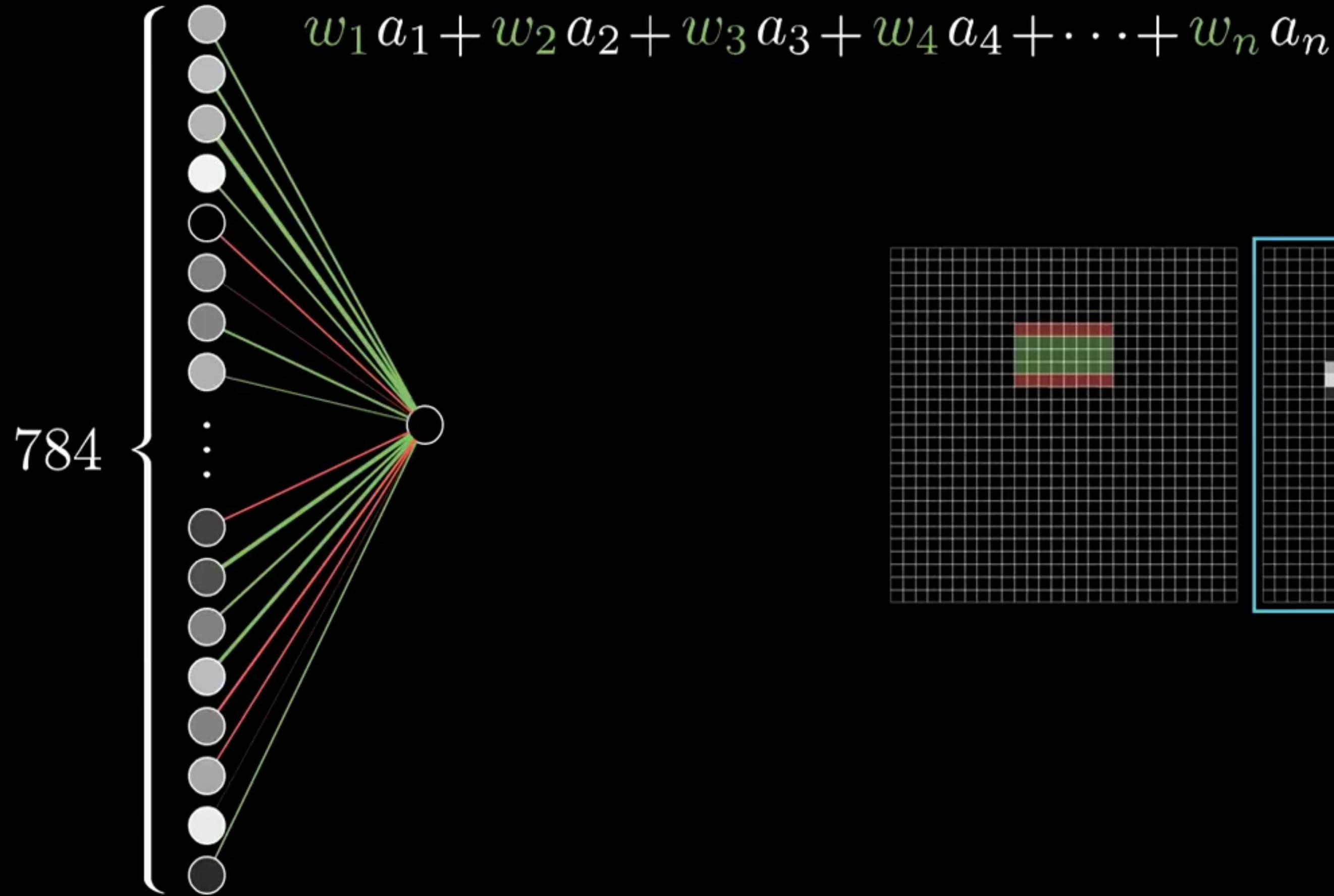


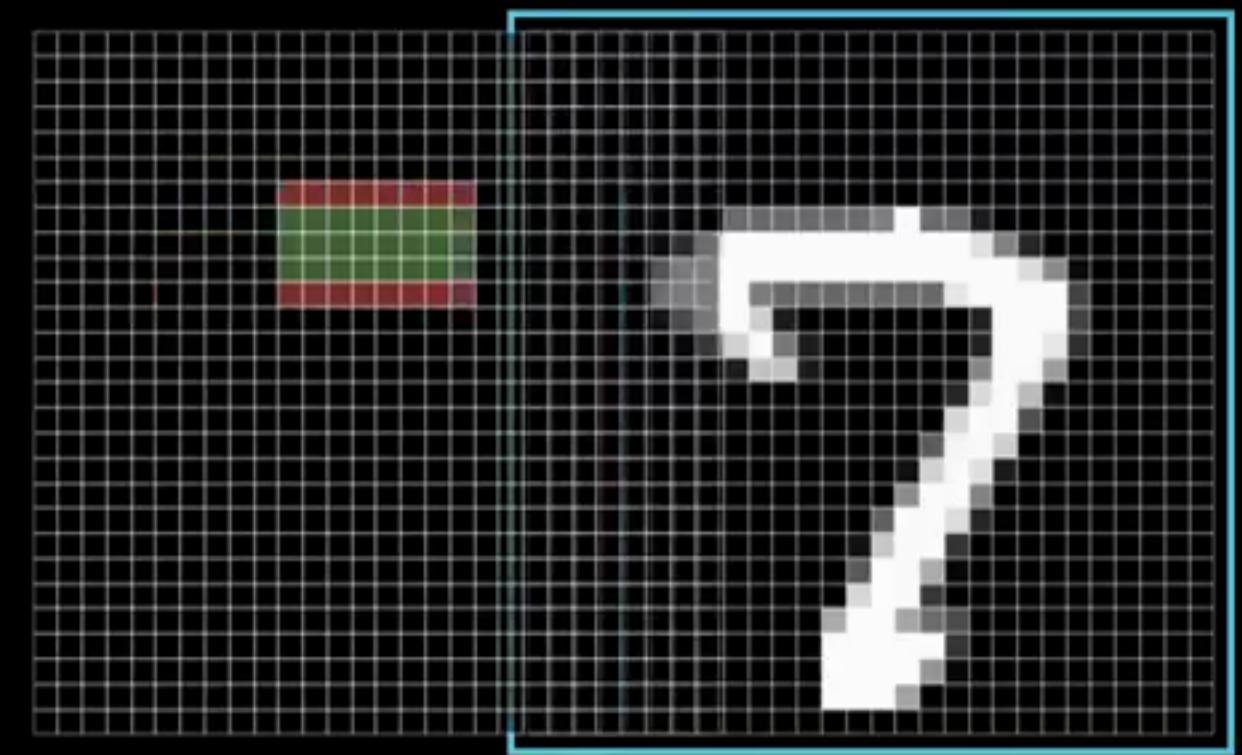
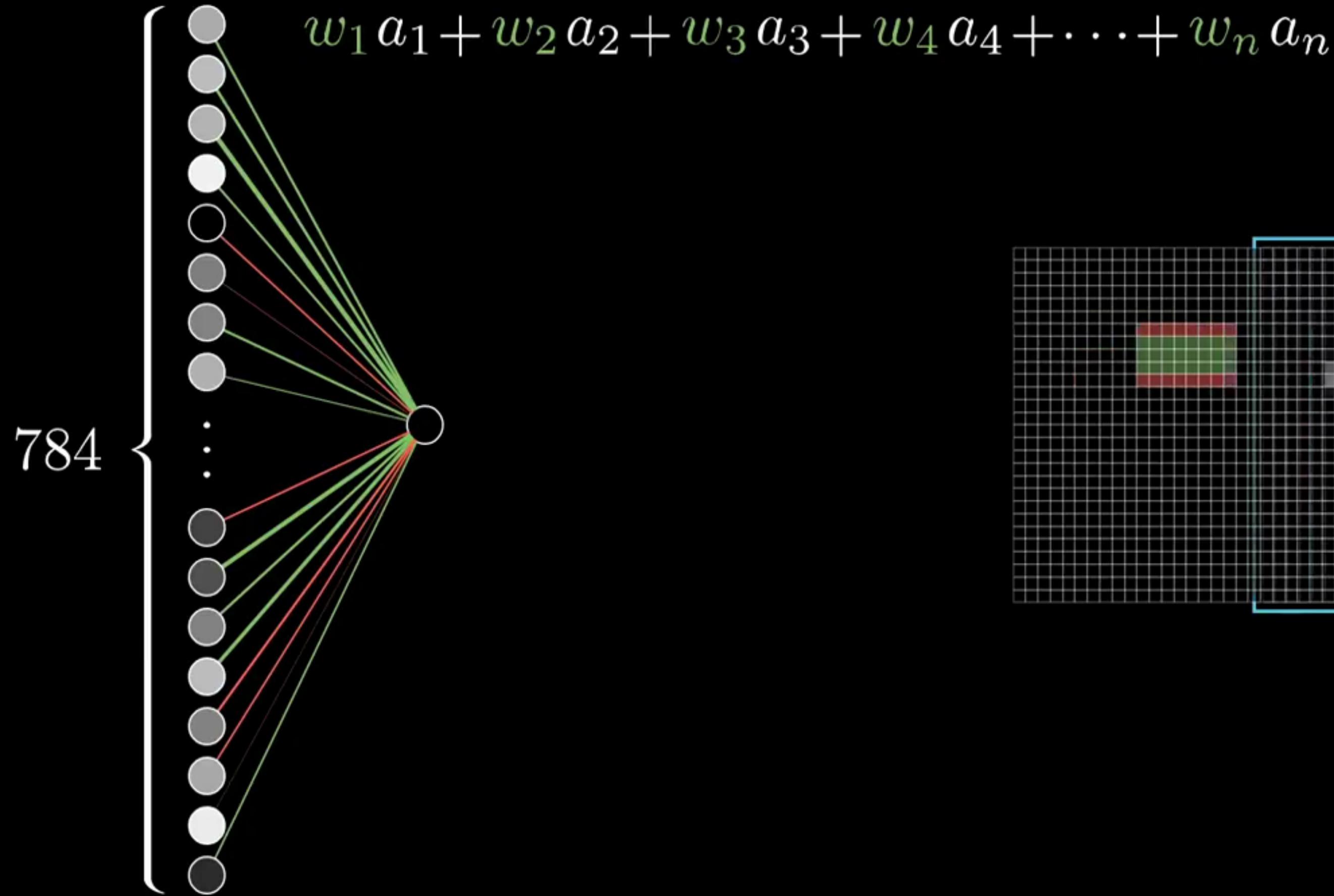


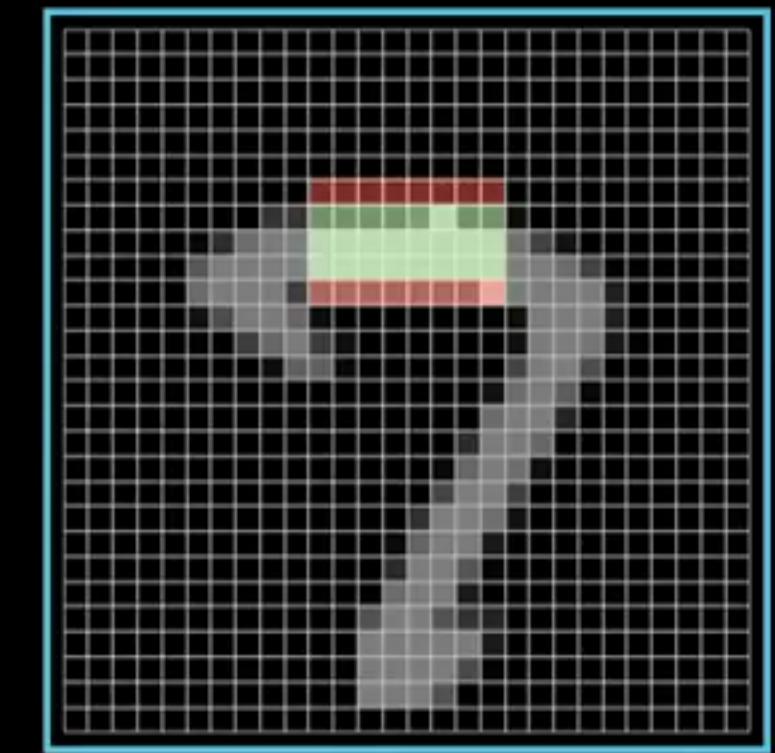
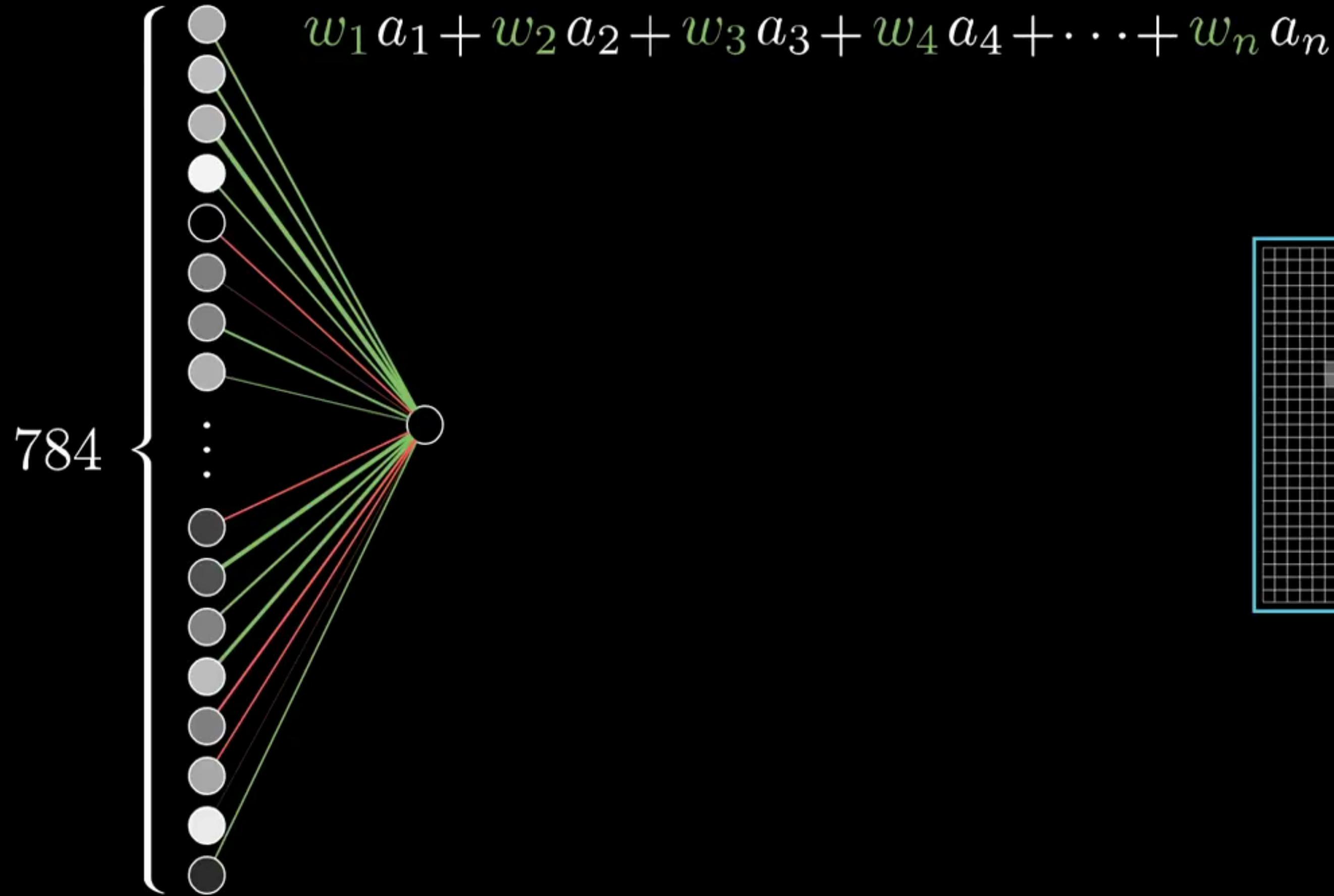




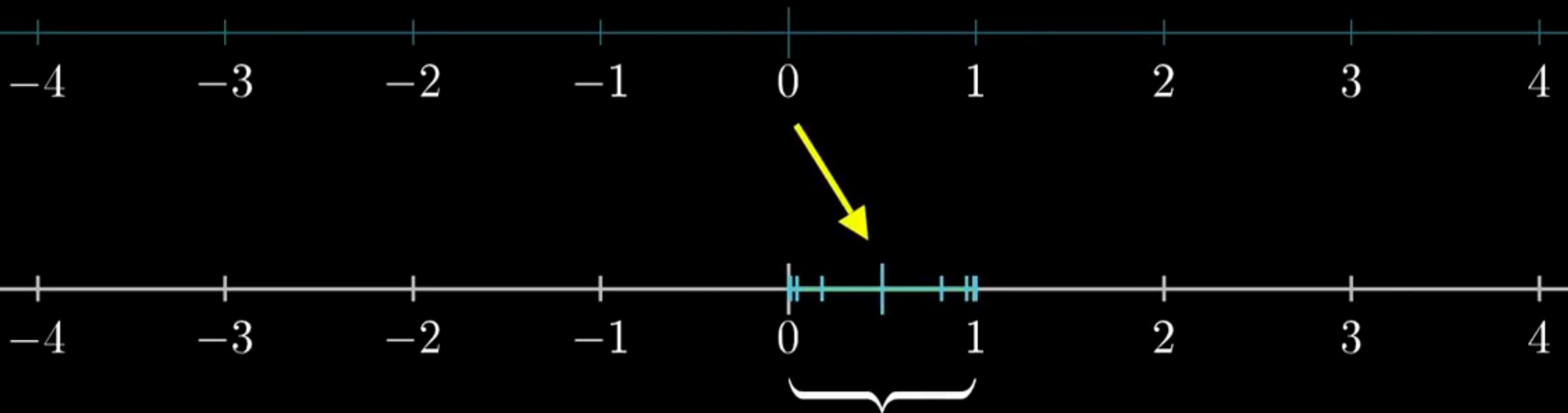








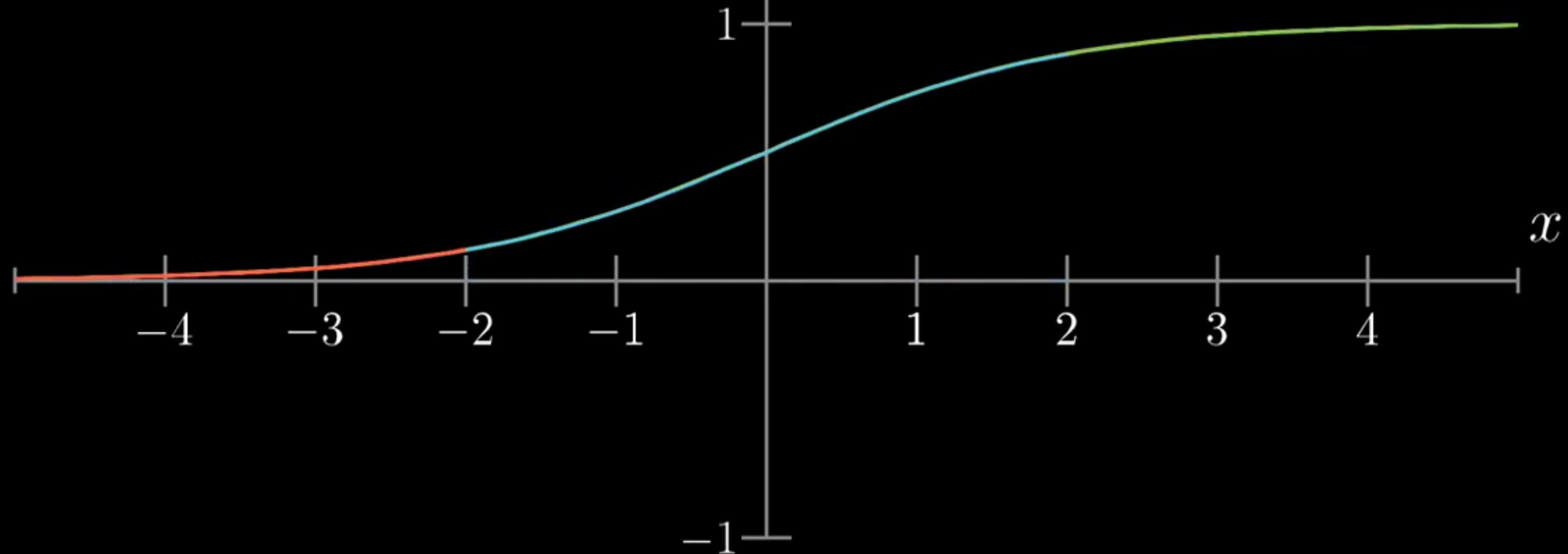
$$w_1 a_1 + w_2 a_2 + w_3 a_3 + w_4 a_4 + \cdots + w_n a_n$$



Activations should be in this range

Sigmoid

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



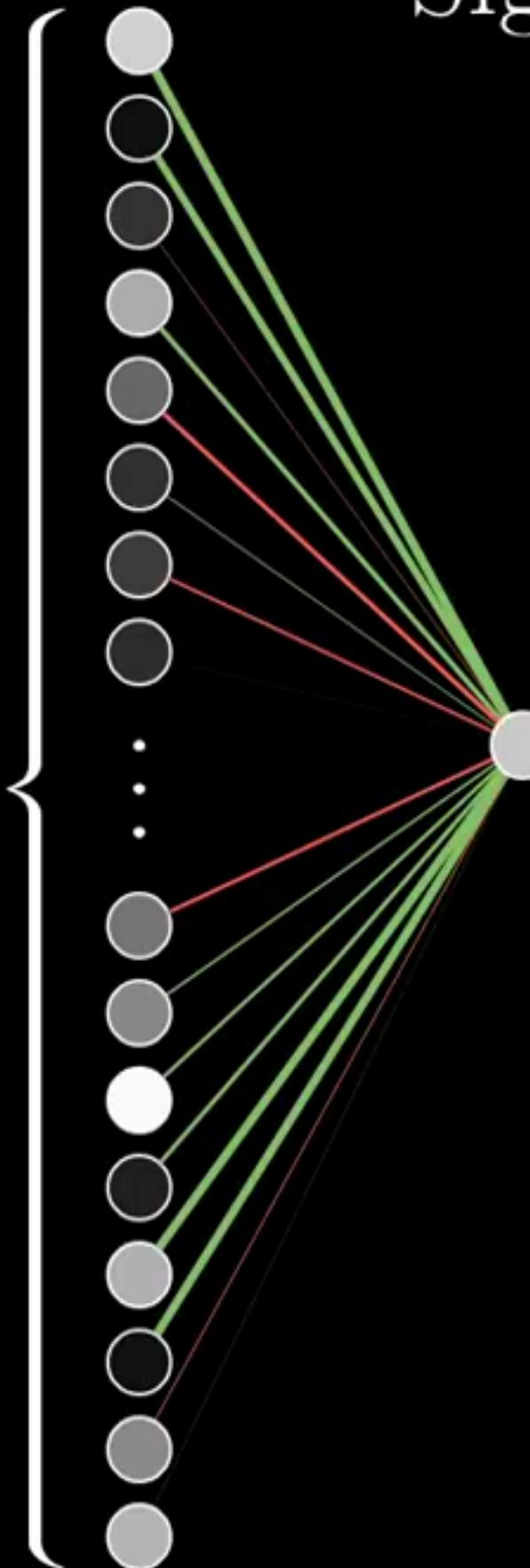
Sigmoid

↓

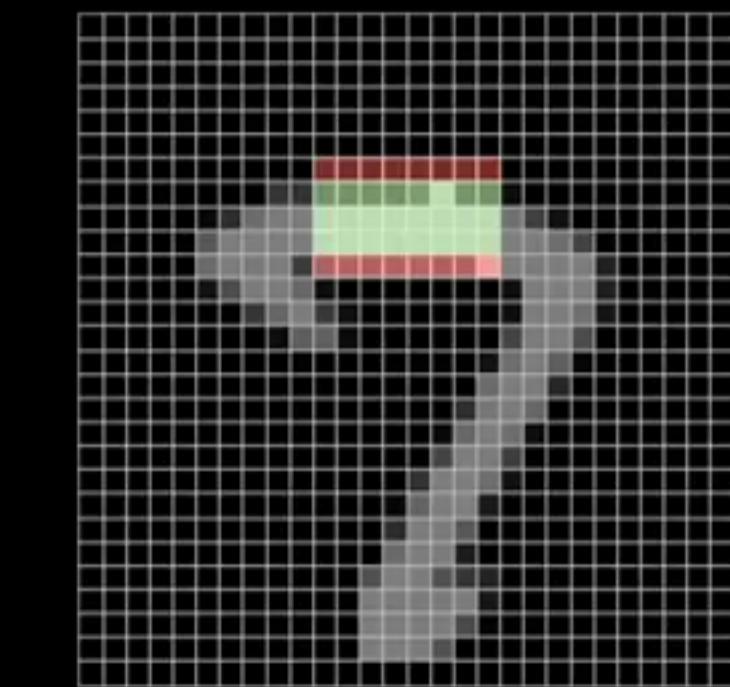
$$\sigma(w_1a_1 + w_2a_2 + w_3a_3 + \cdots + w_na_n[-10])$$

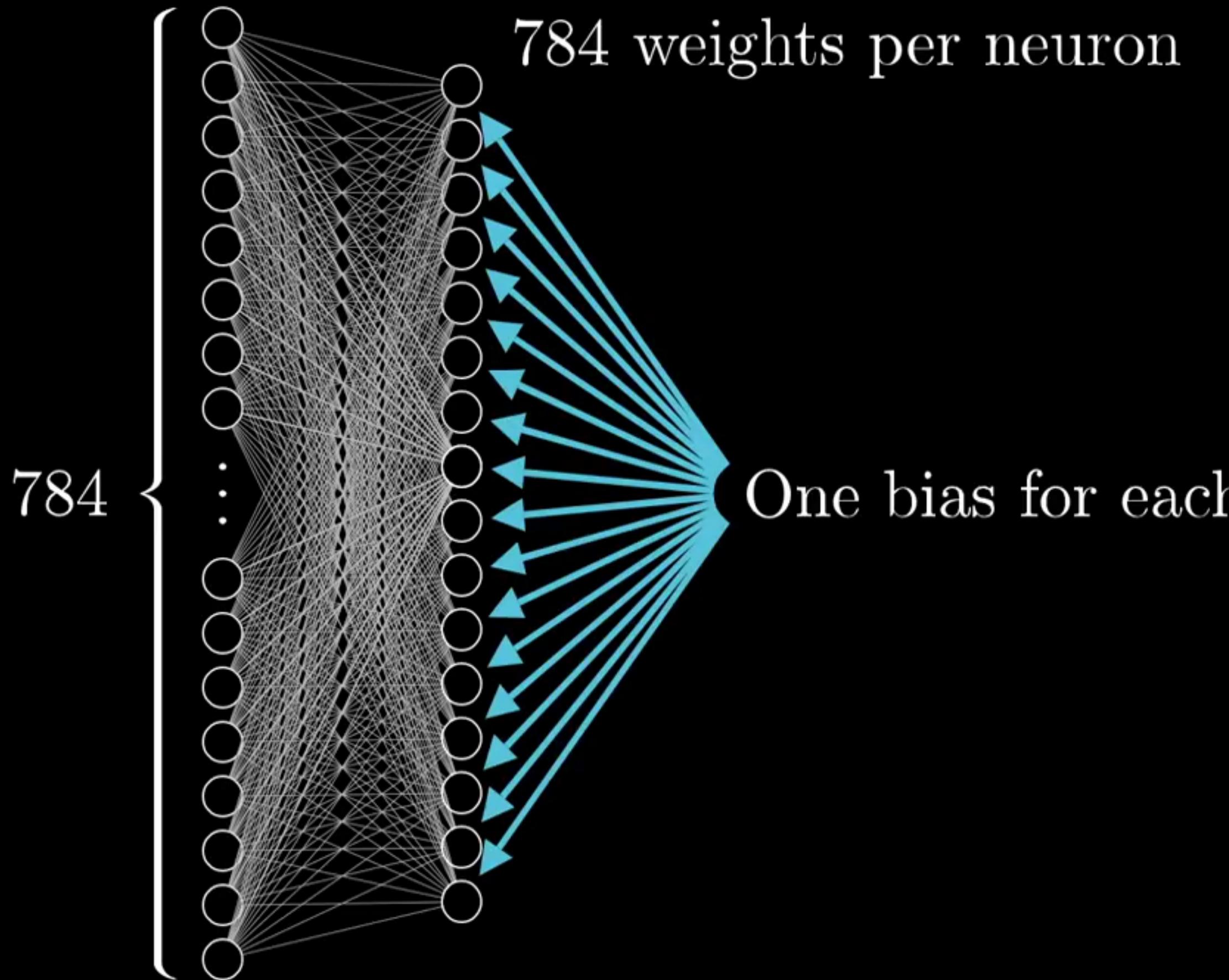
“bias”

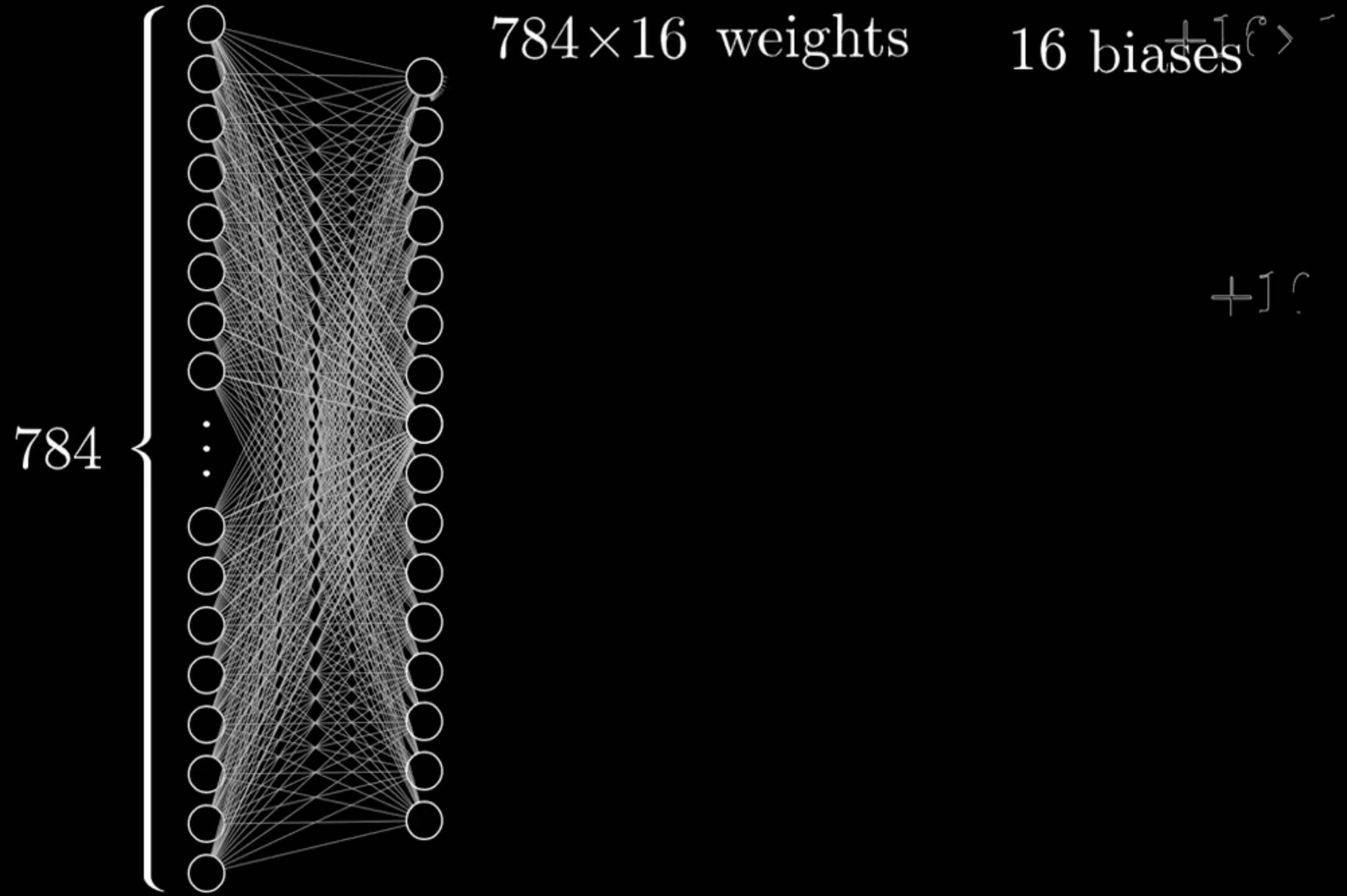
784

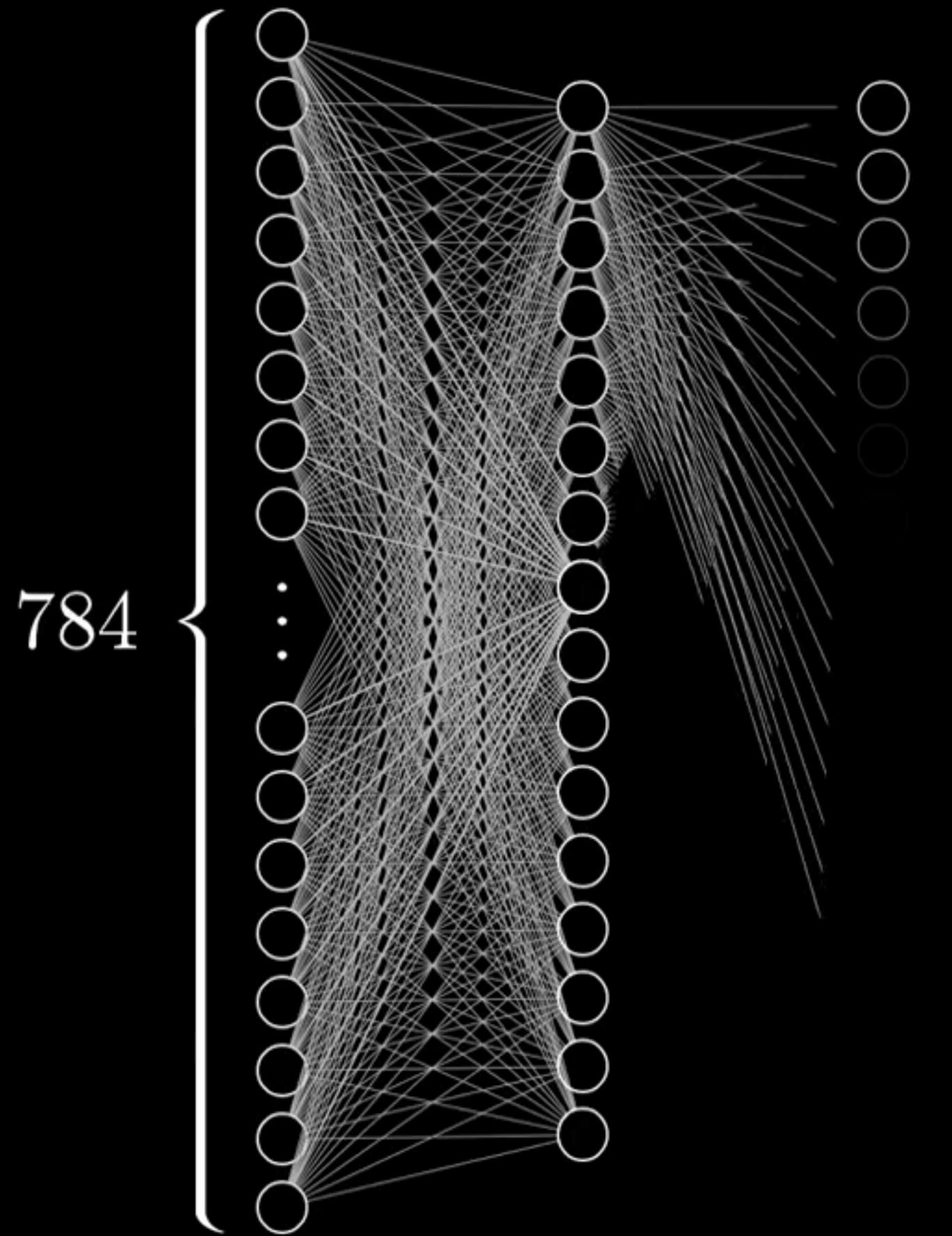


Only activate meaningfully
when weighted sum > 10



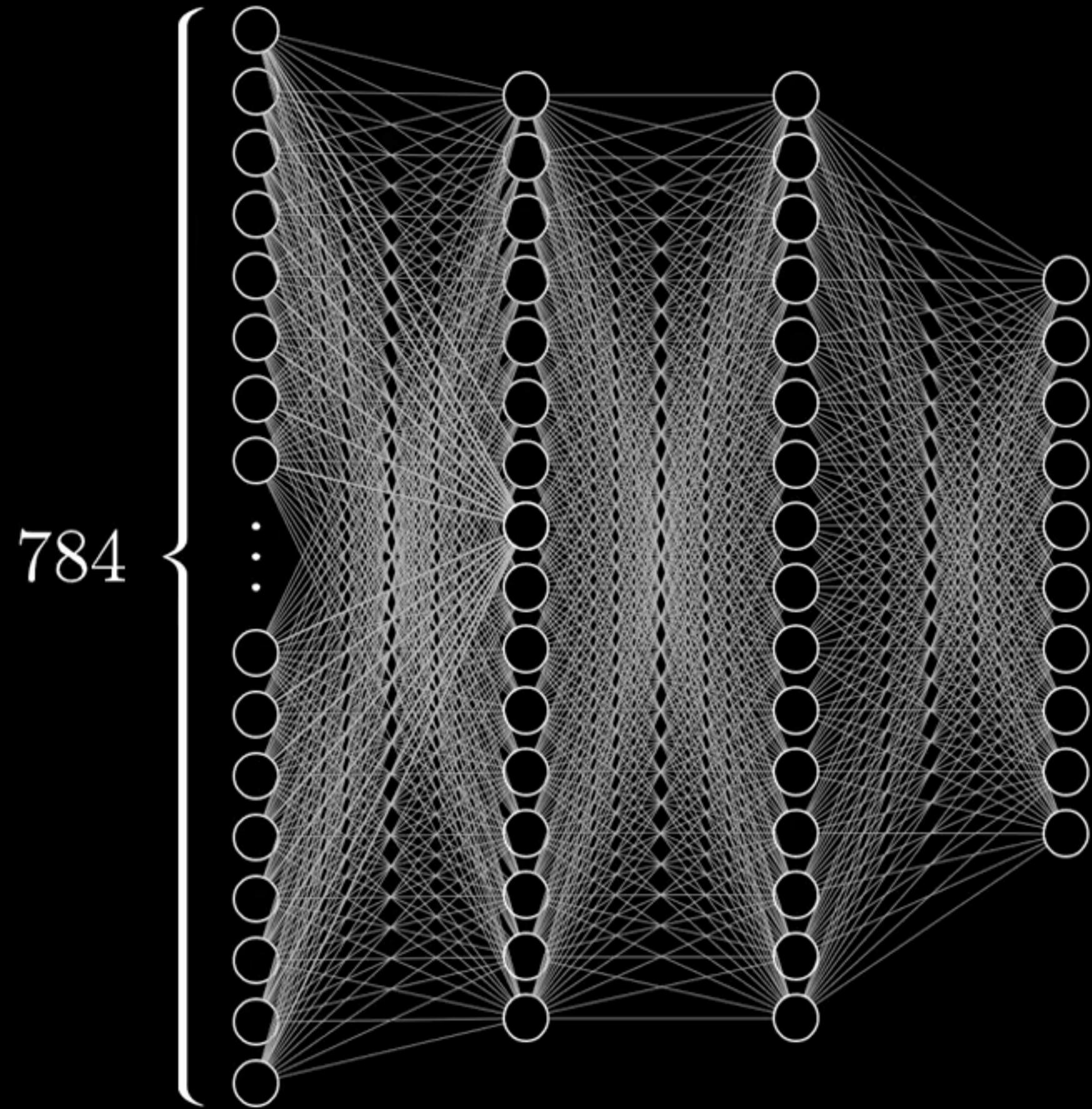






$784 \times 16 + 16 \times 16 + 16 \times 10$
weights

$16 + 16 + 10$
biases



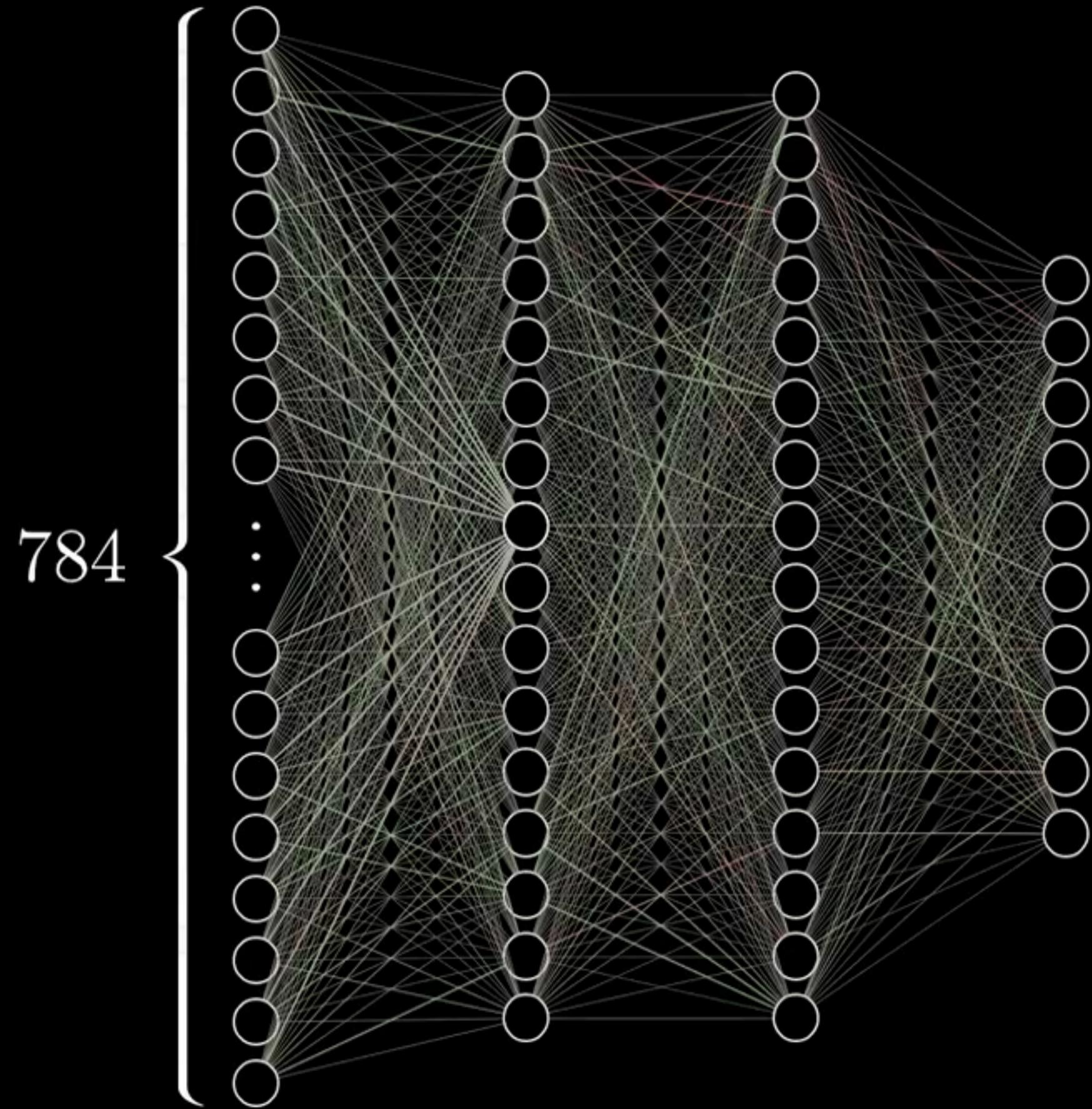
$784 \times 16 + 16 \times 16 + 16 \times 10$

weights

16 + 16 + 10

biases

13,002



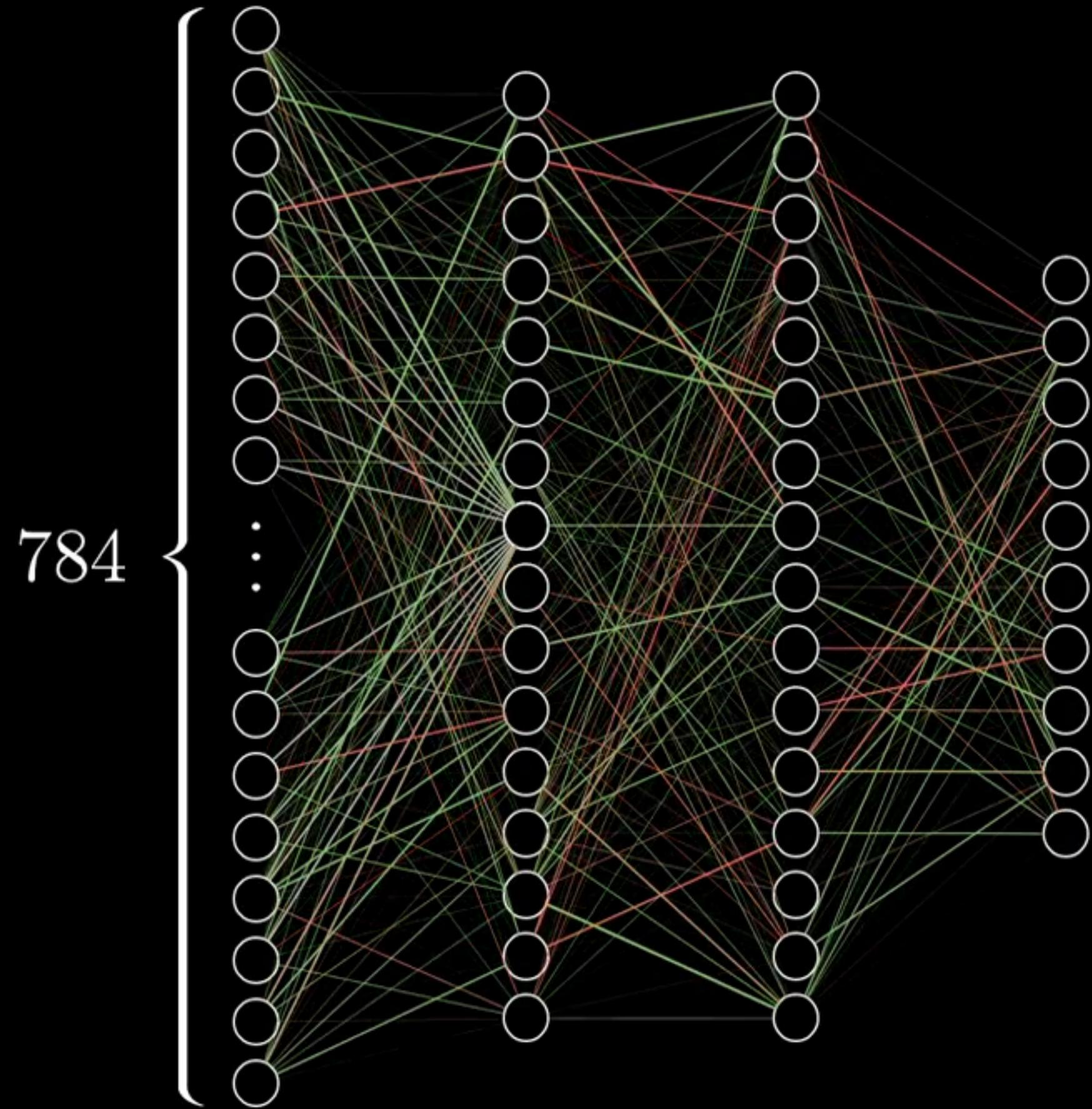
$$784 \times 16 + 16 \times 16 + 16 \times 10$$

weights

$$16 + 16 + 10$$

biases

13,002



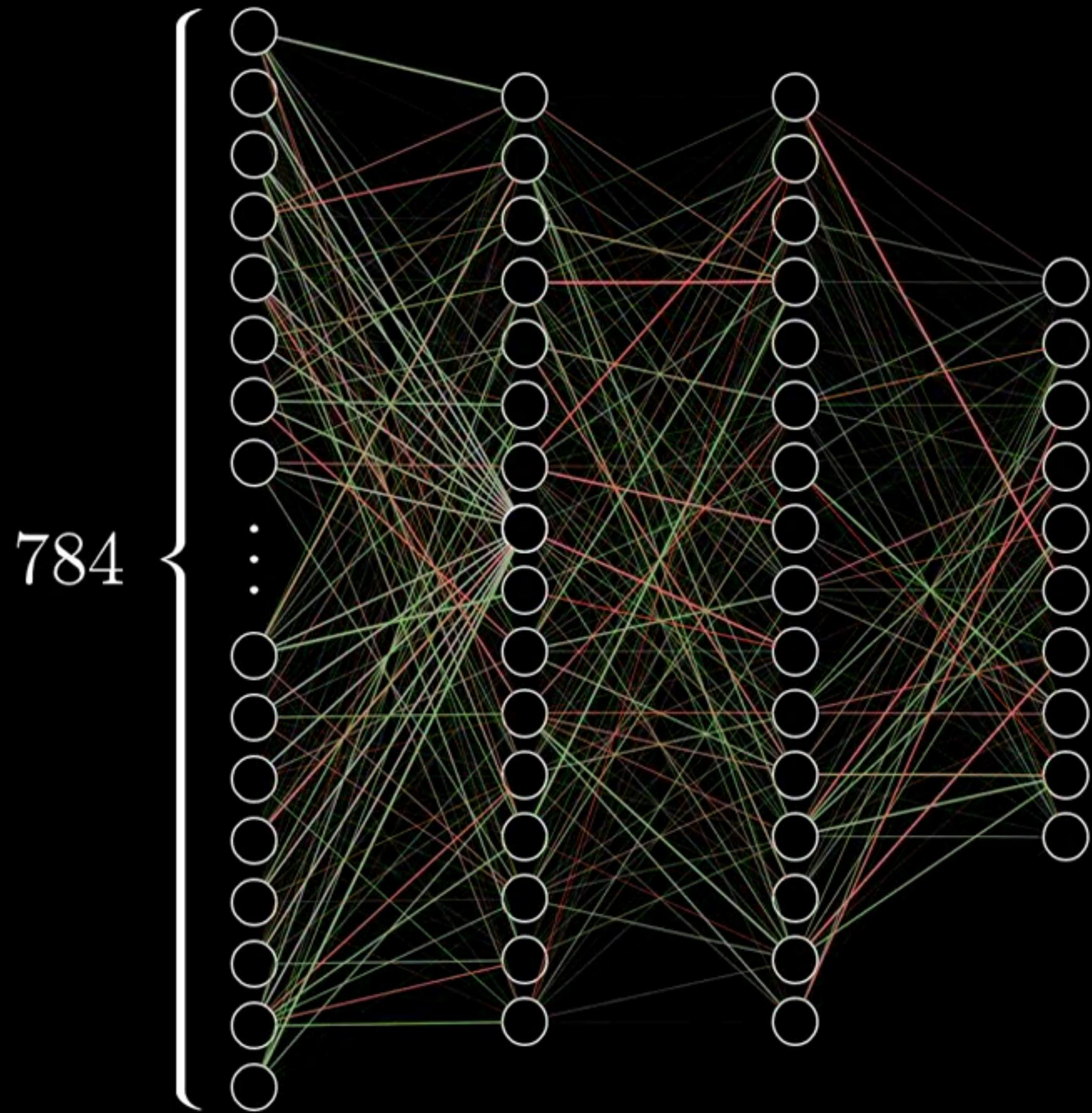
$784 \times 16 + 16 \times 16 + 16 \times 10$

weights

16 + 16 + 10

biases

13,002



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

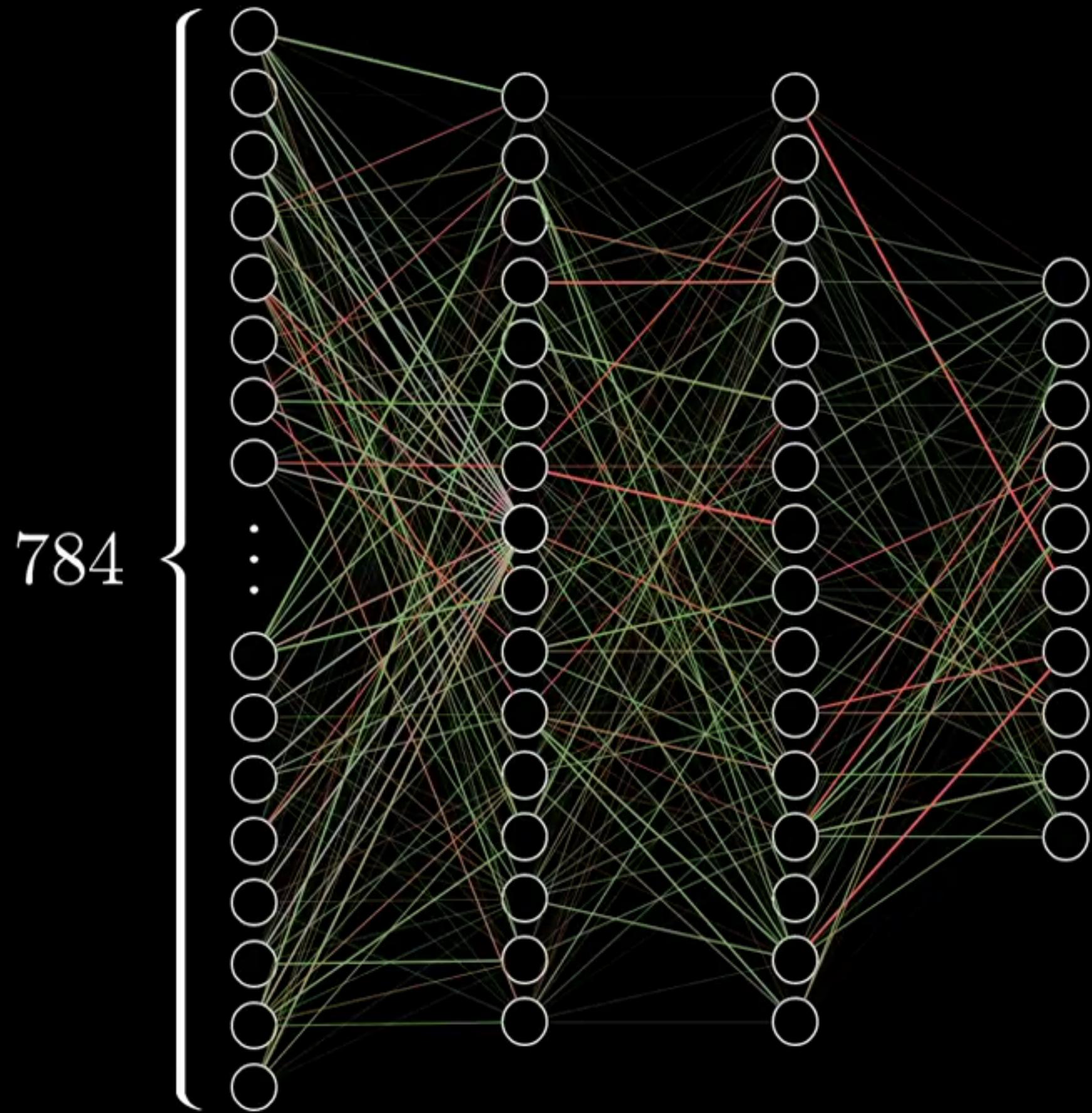
weights

$$16 + 16 + 10$$

biases

13,002

Learning → Finding the right
weights and biases



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

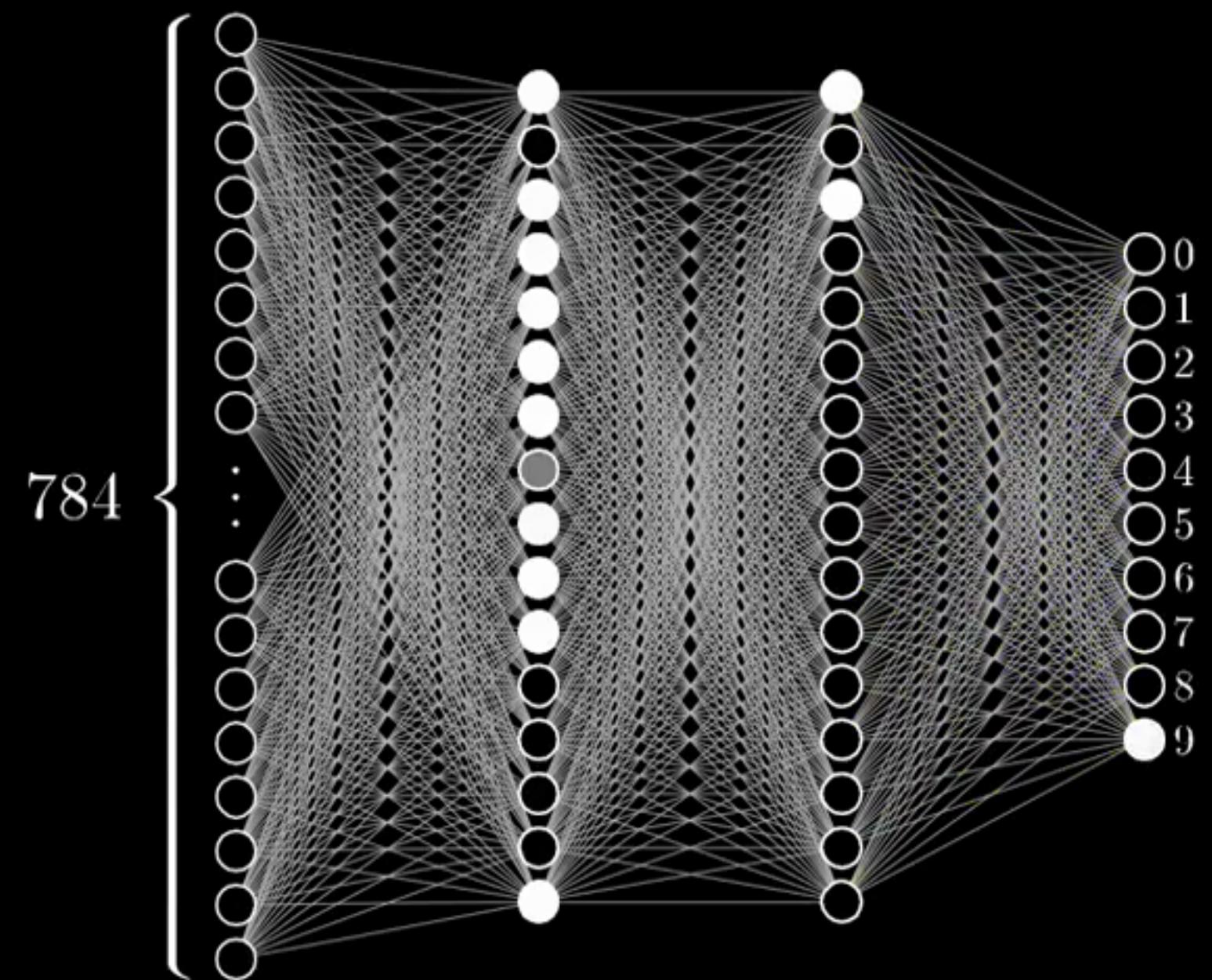
weights

$$16 + 16 + 10$$

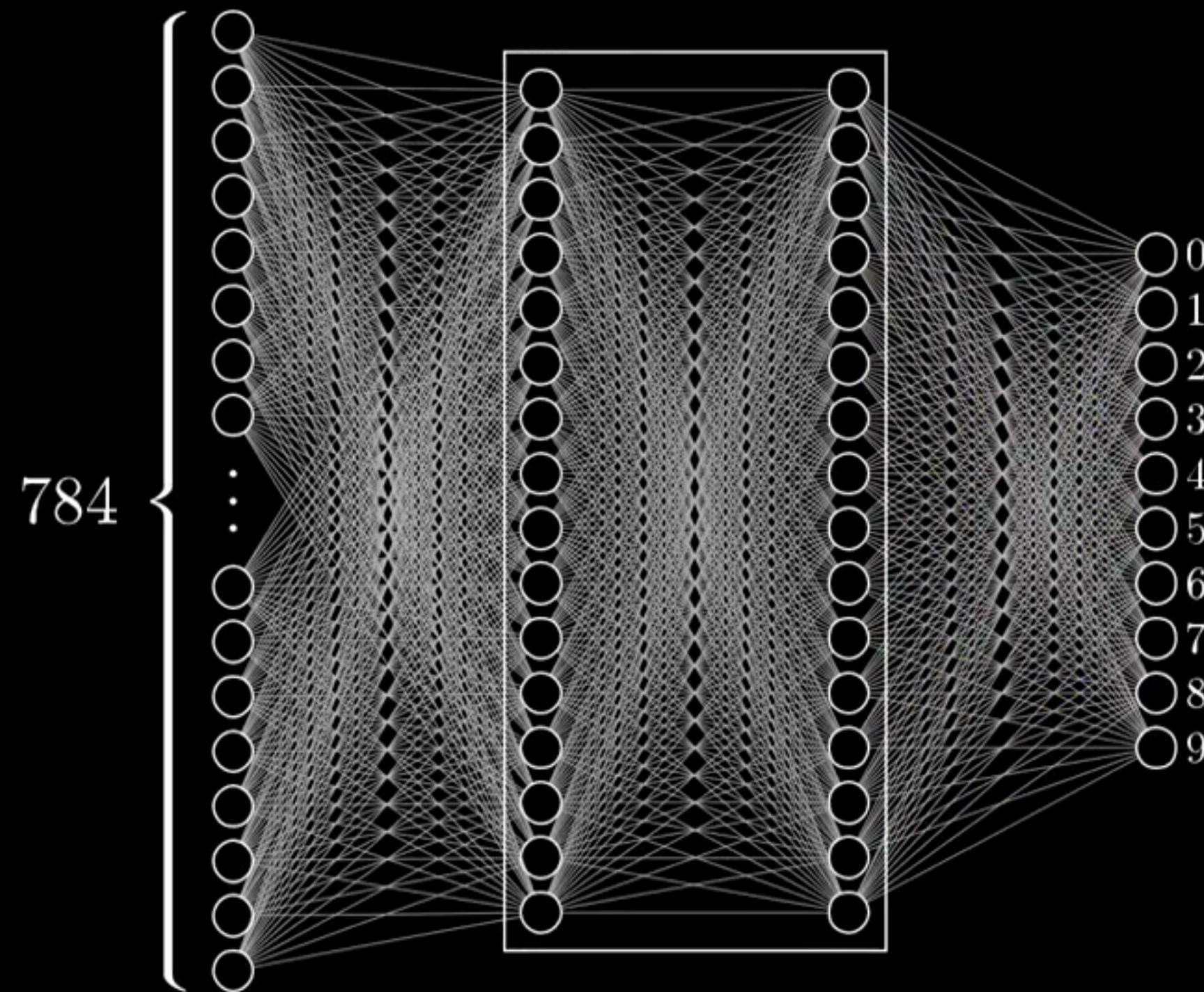
biases

13,002

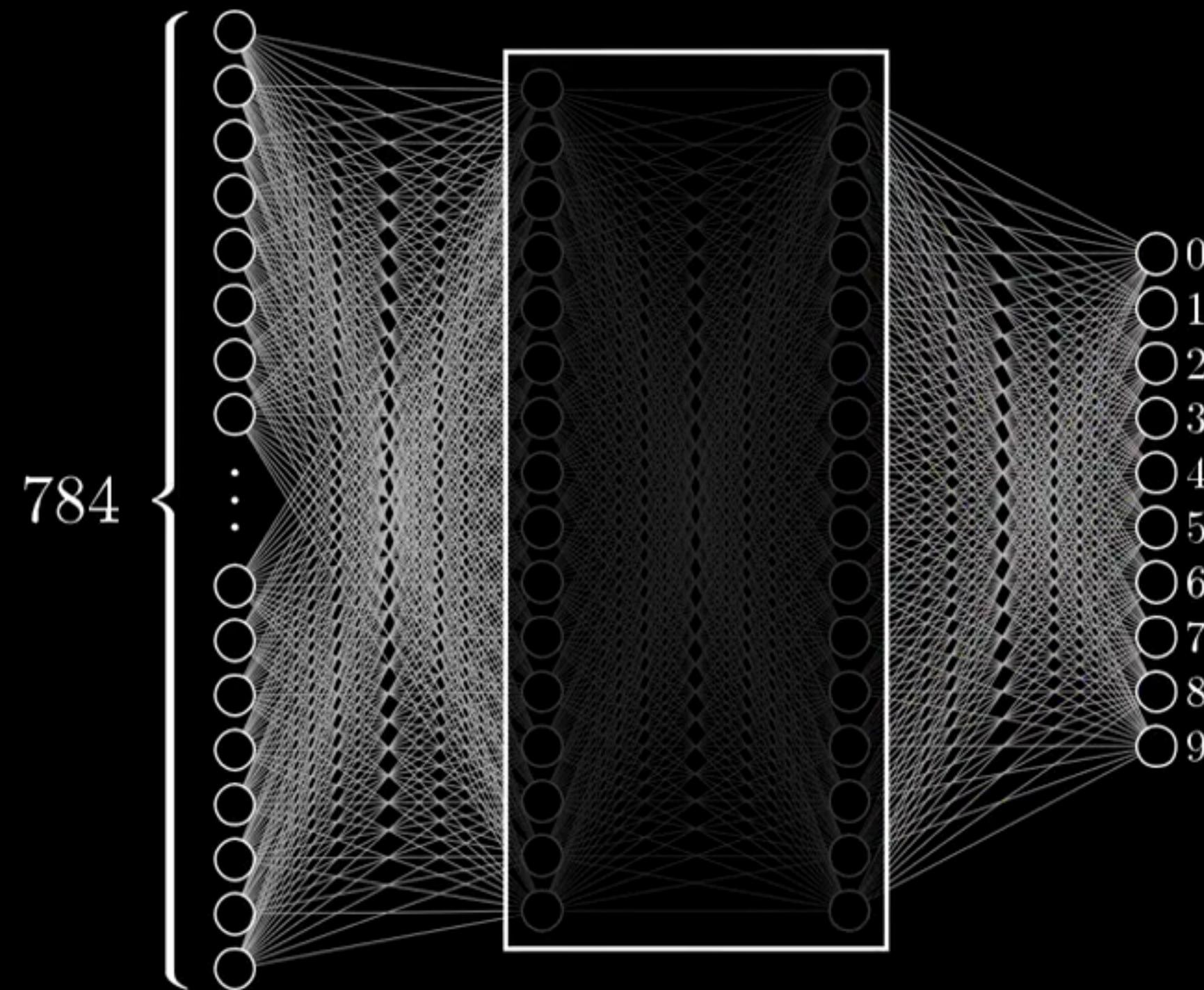
Learning → Finding the right
weights and biases

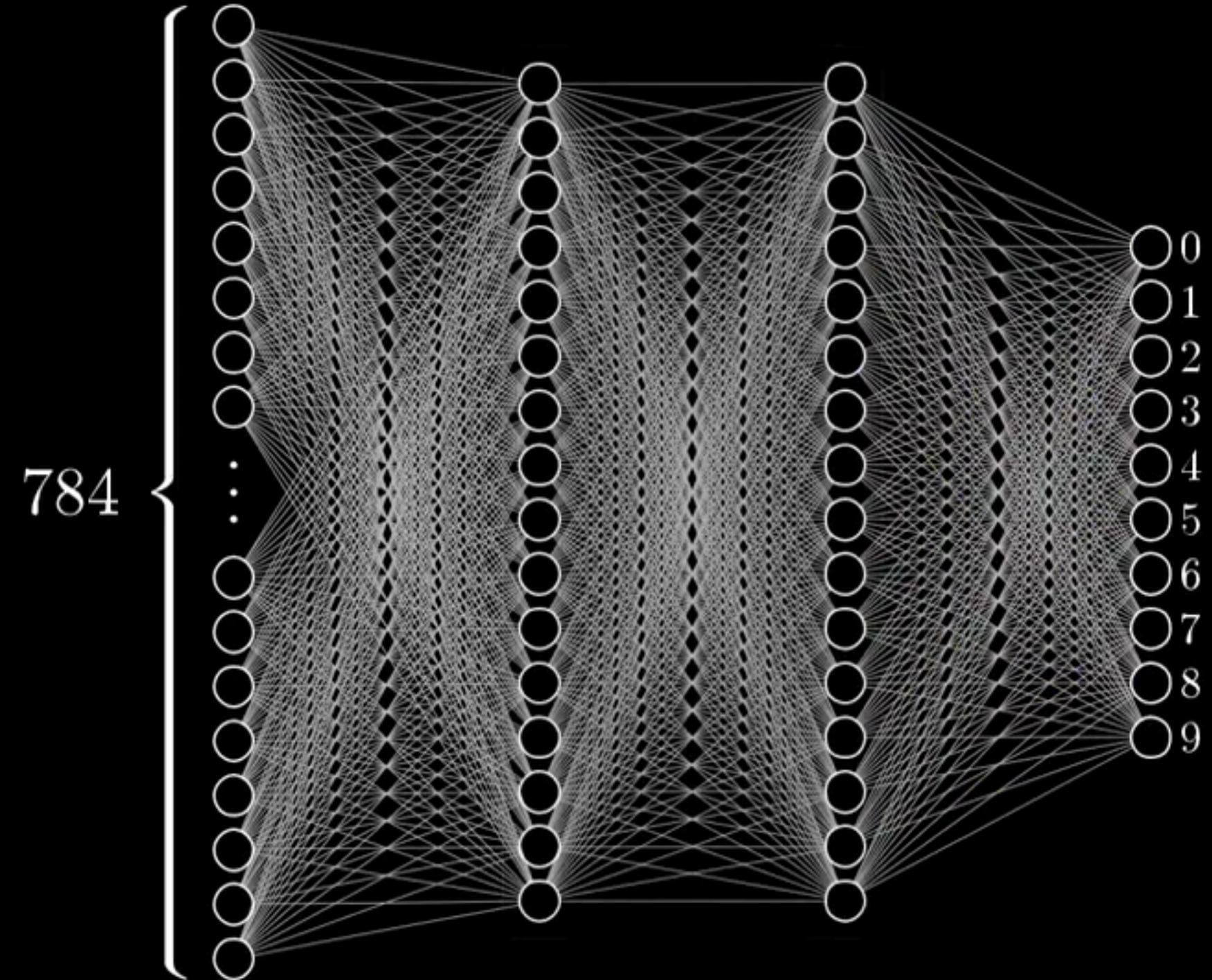


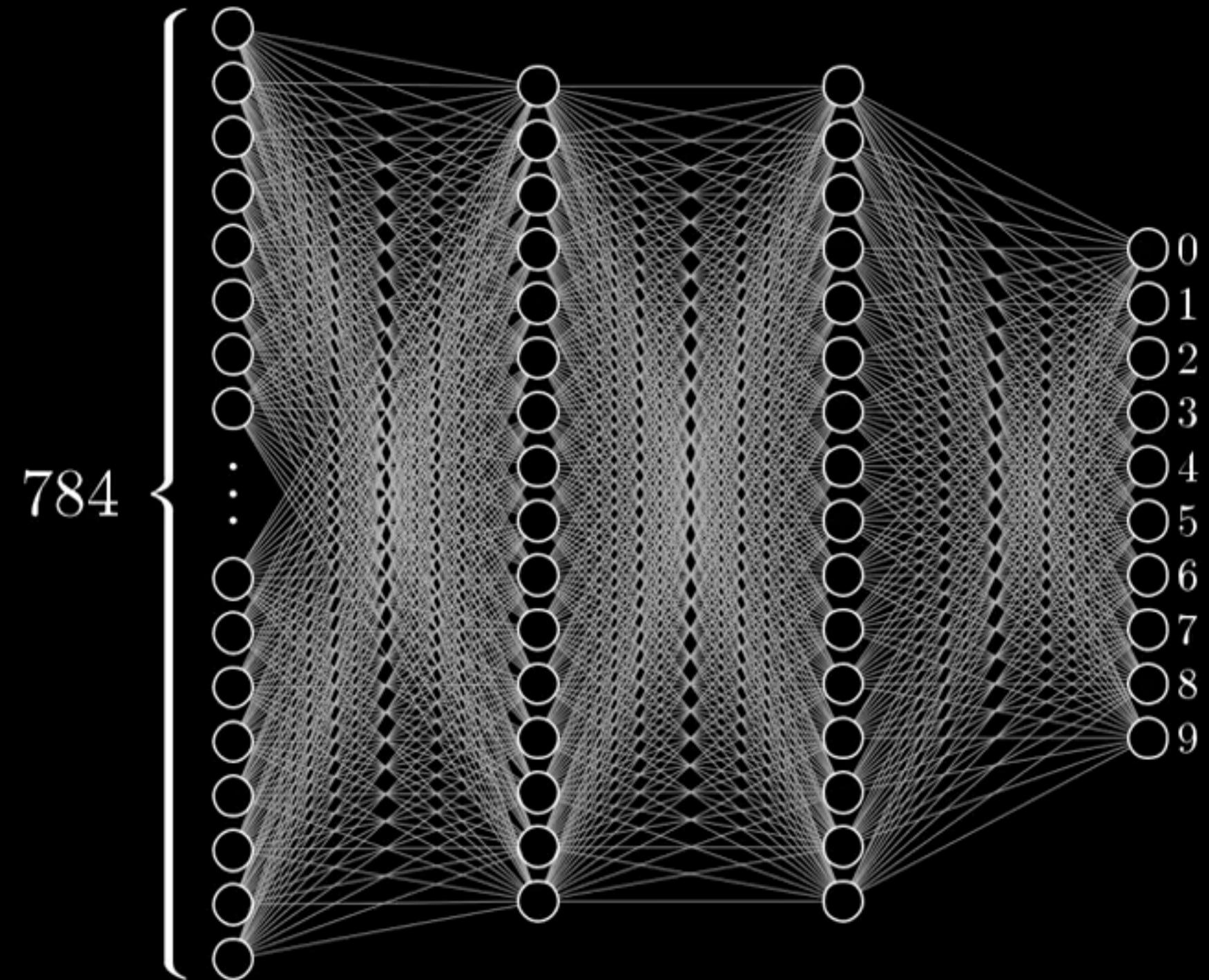
...rather than treating this as a black box

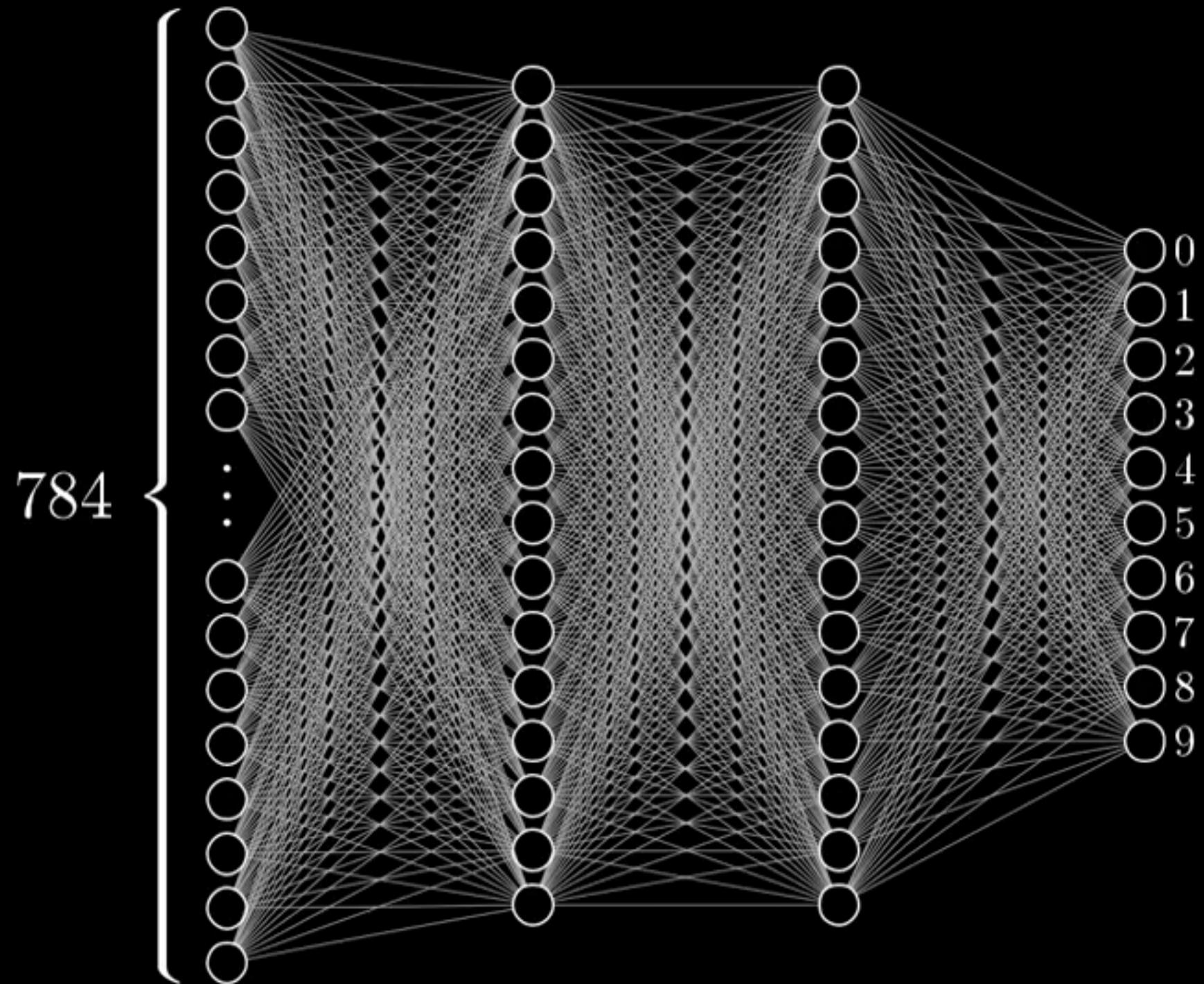


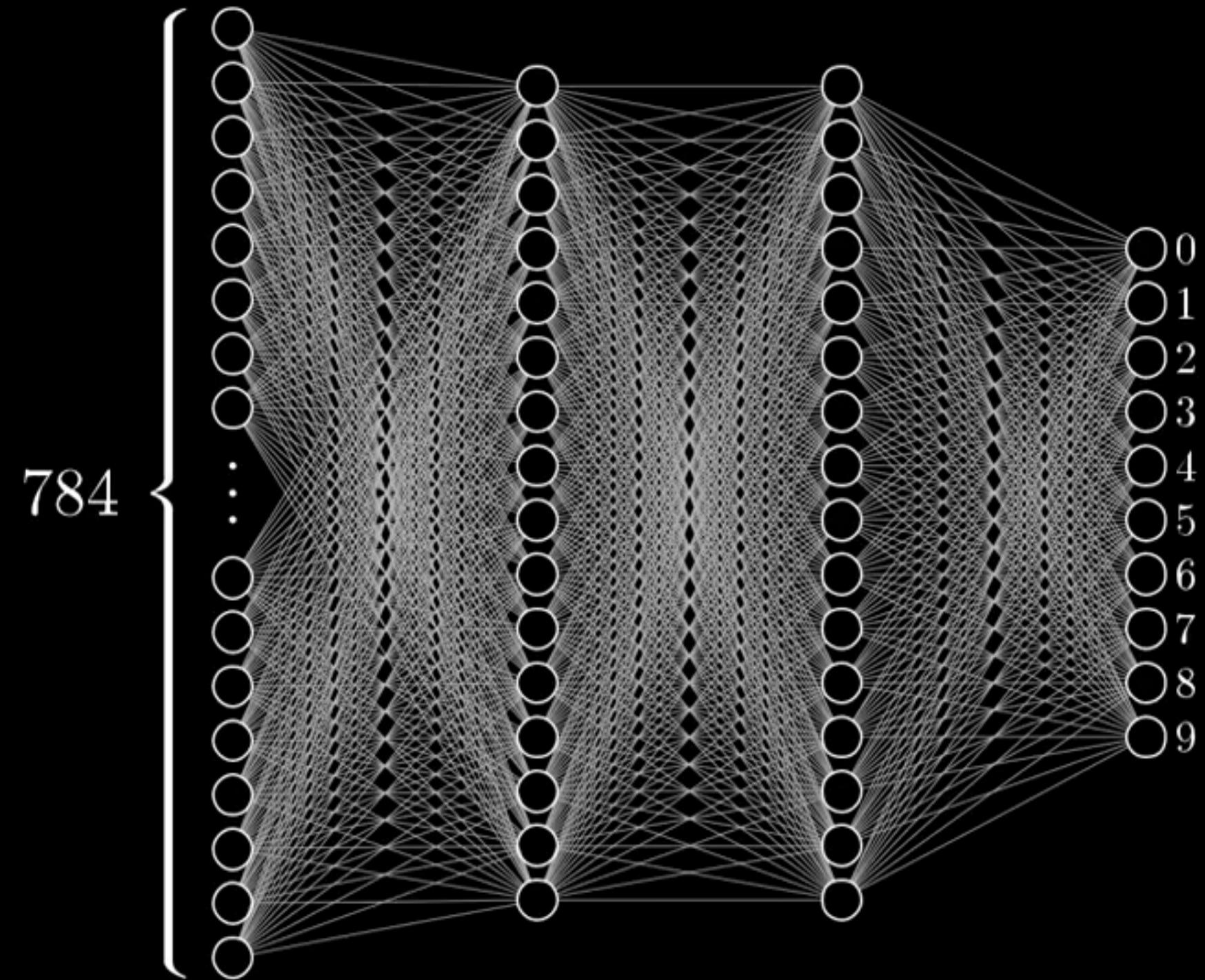
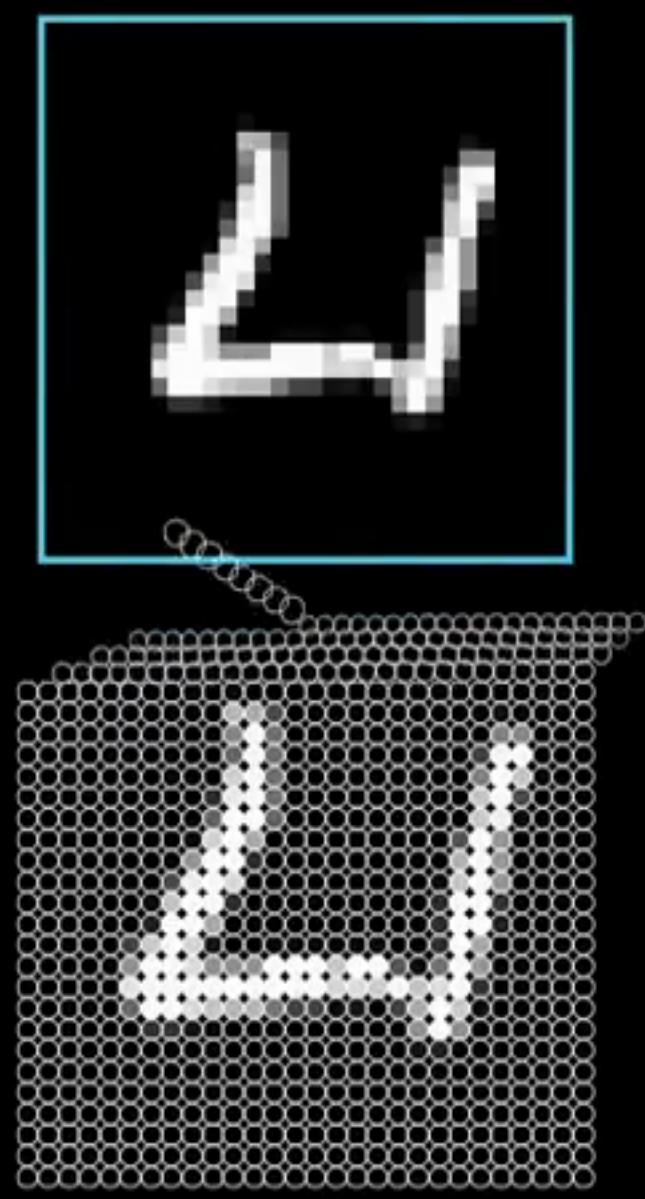
...rather than treating this as a black box

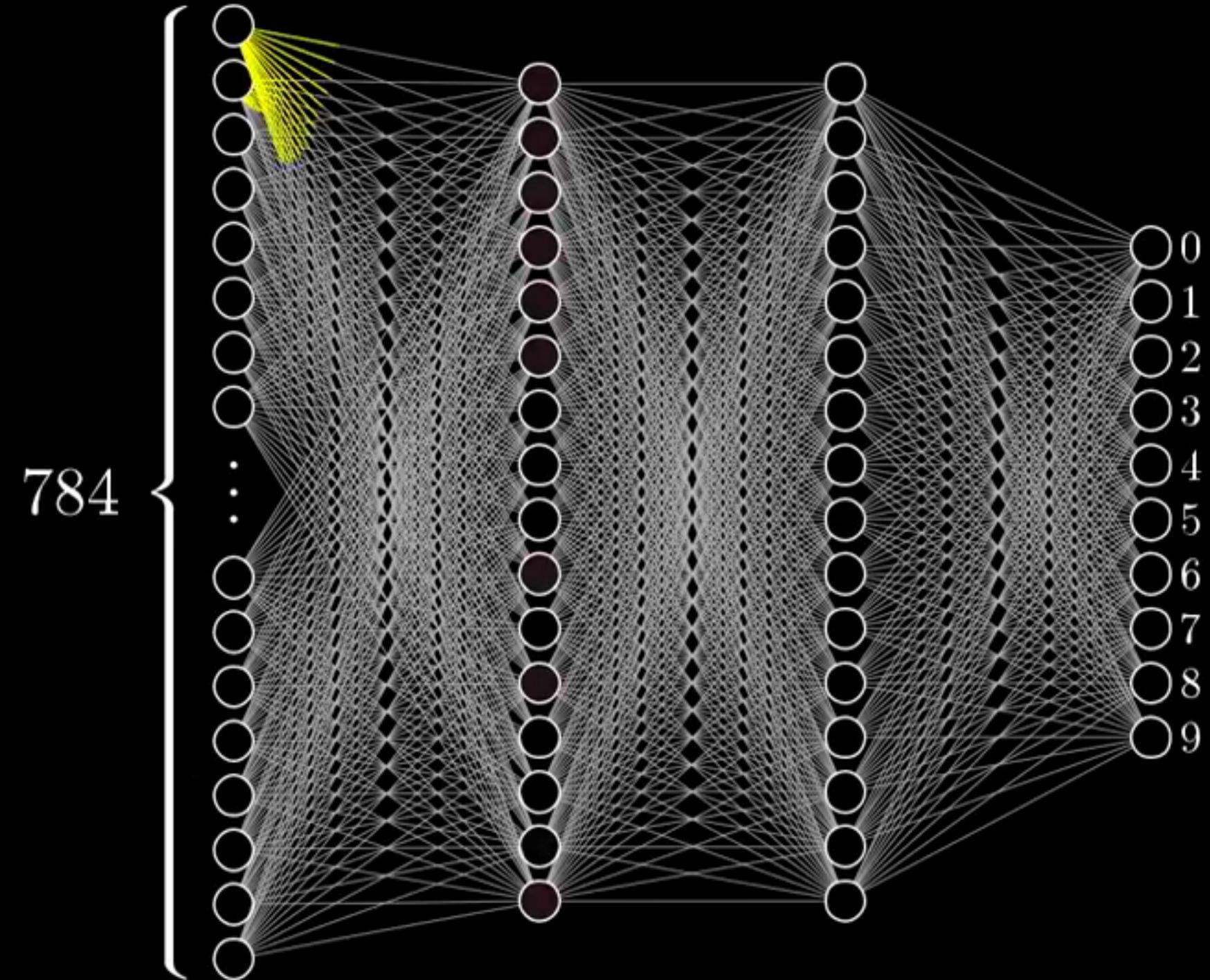


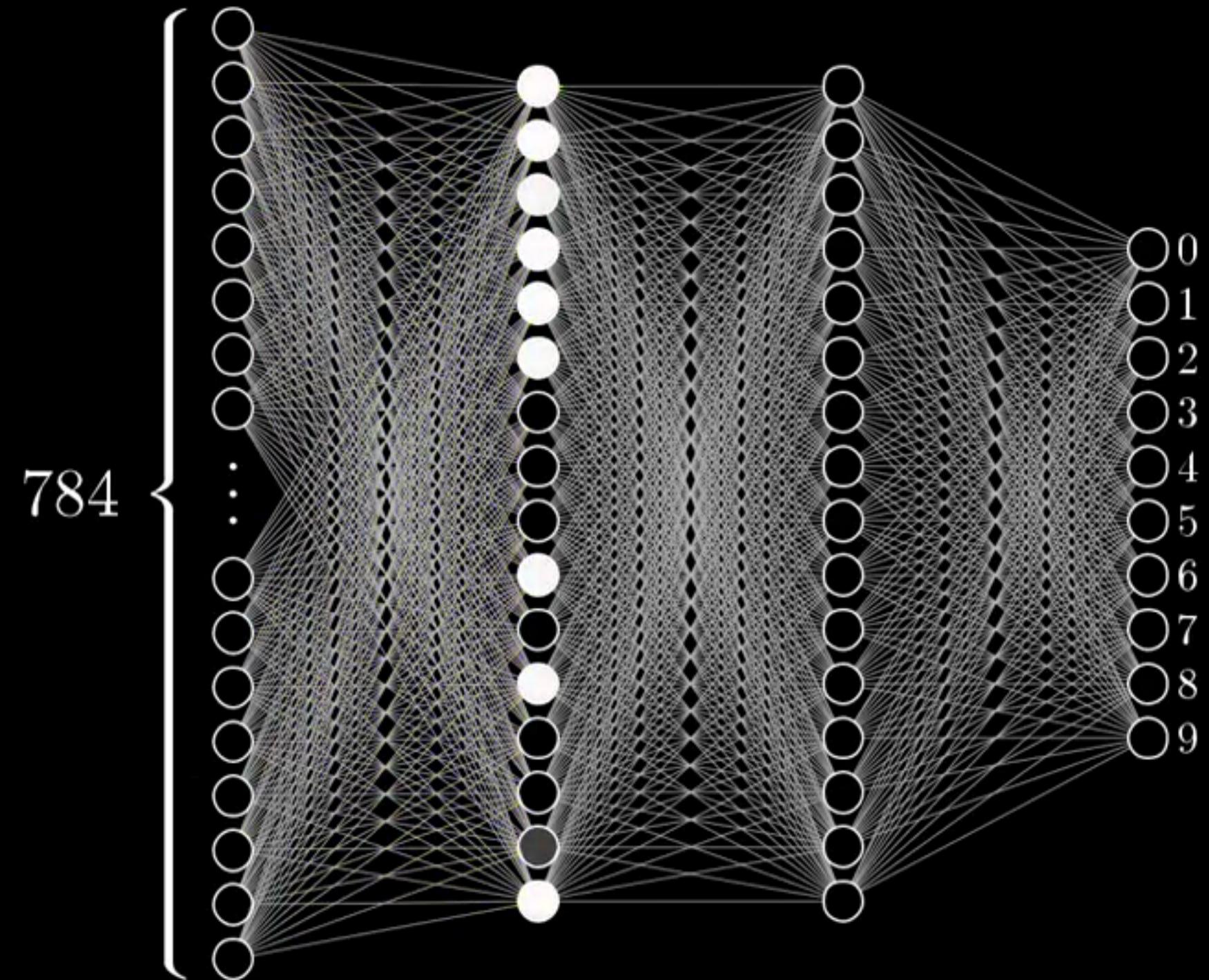


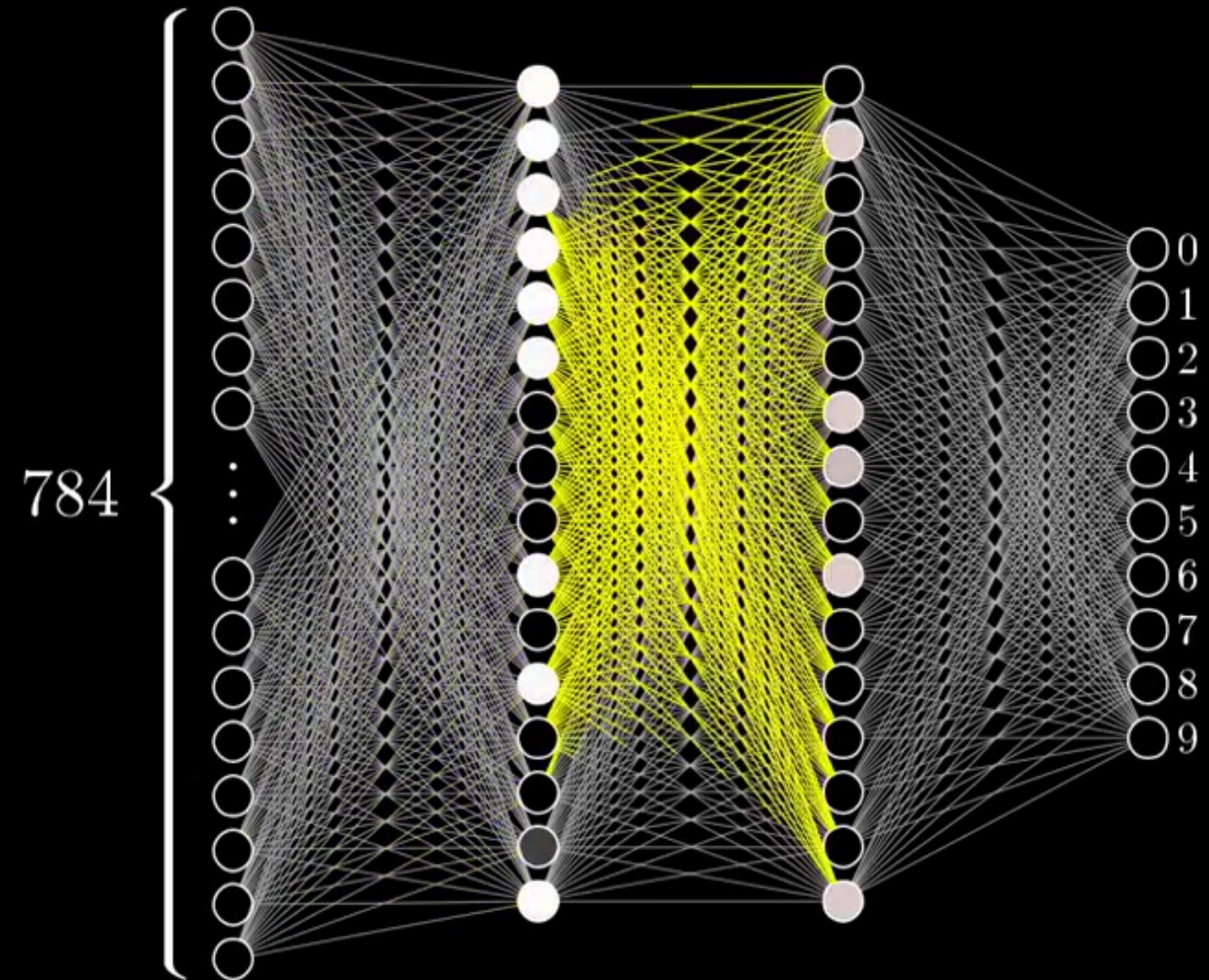






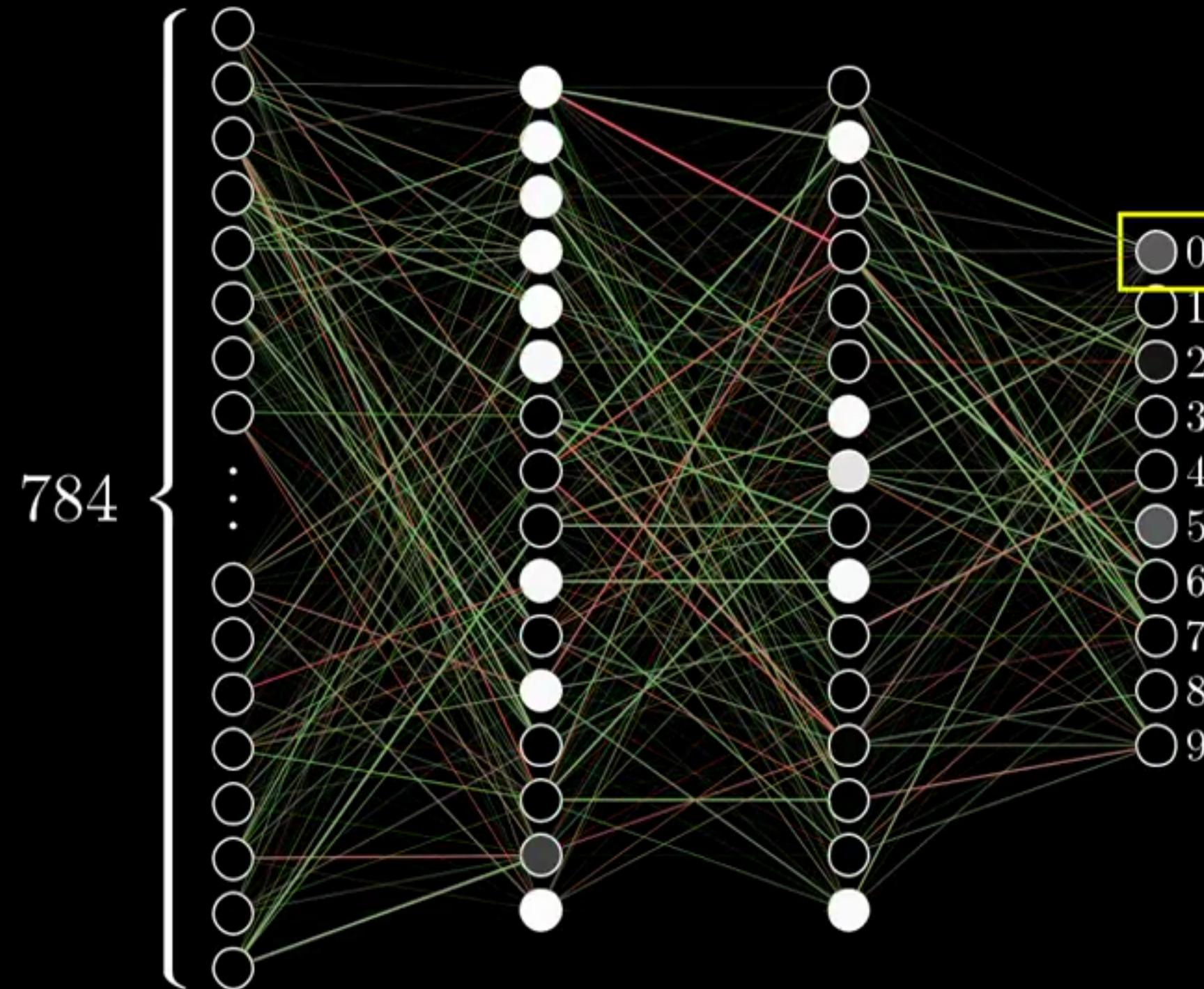




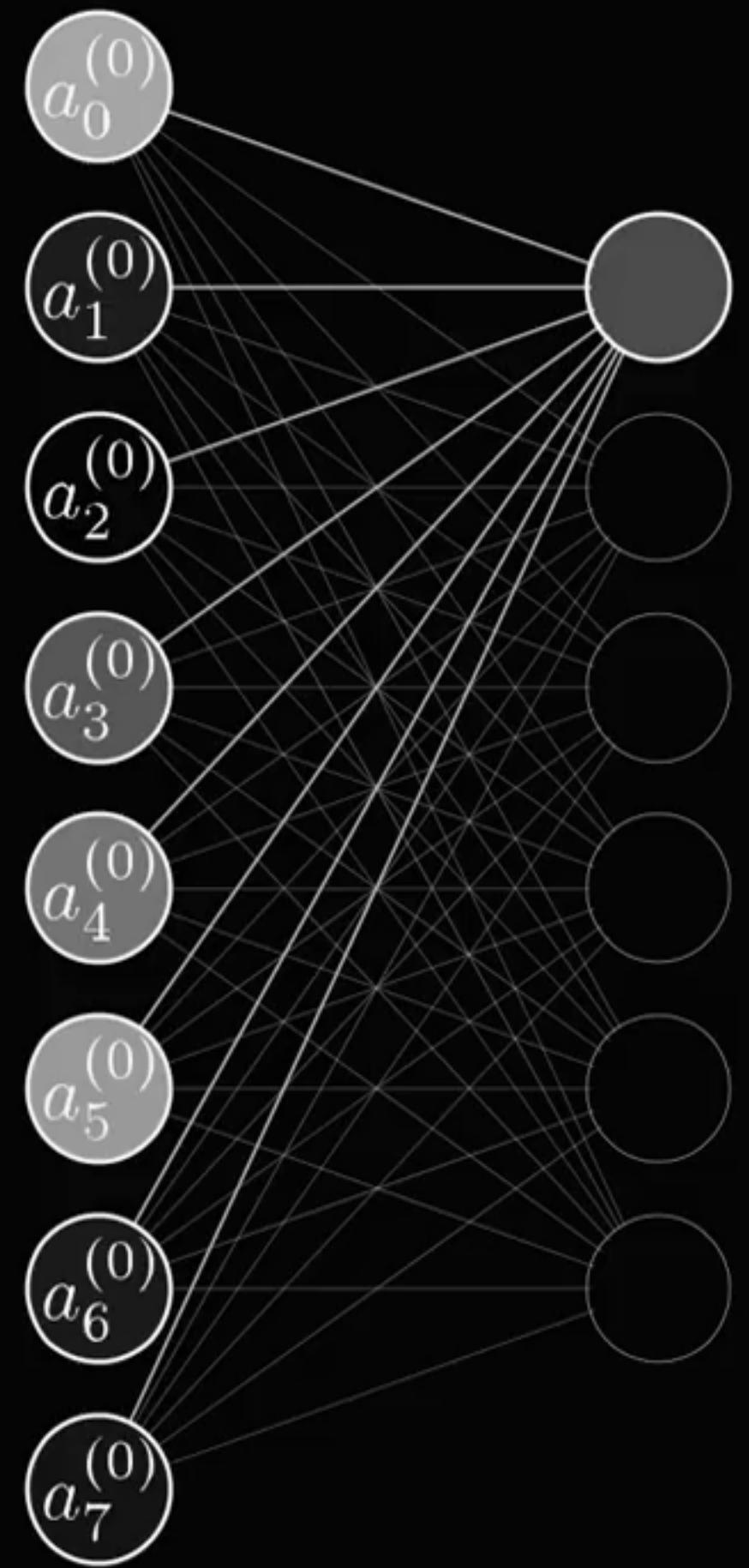


What weights are used here?

What are they doing?



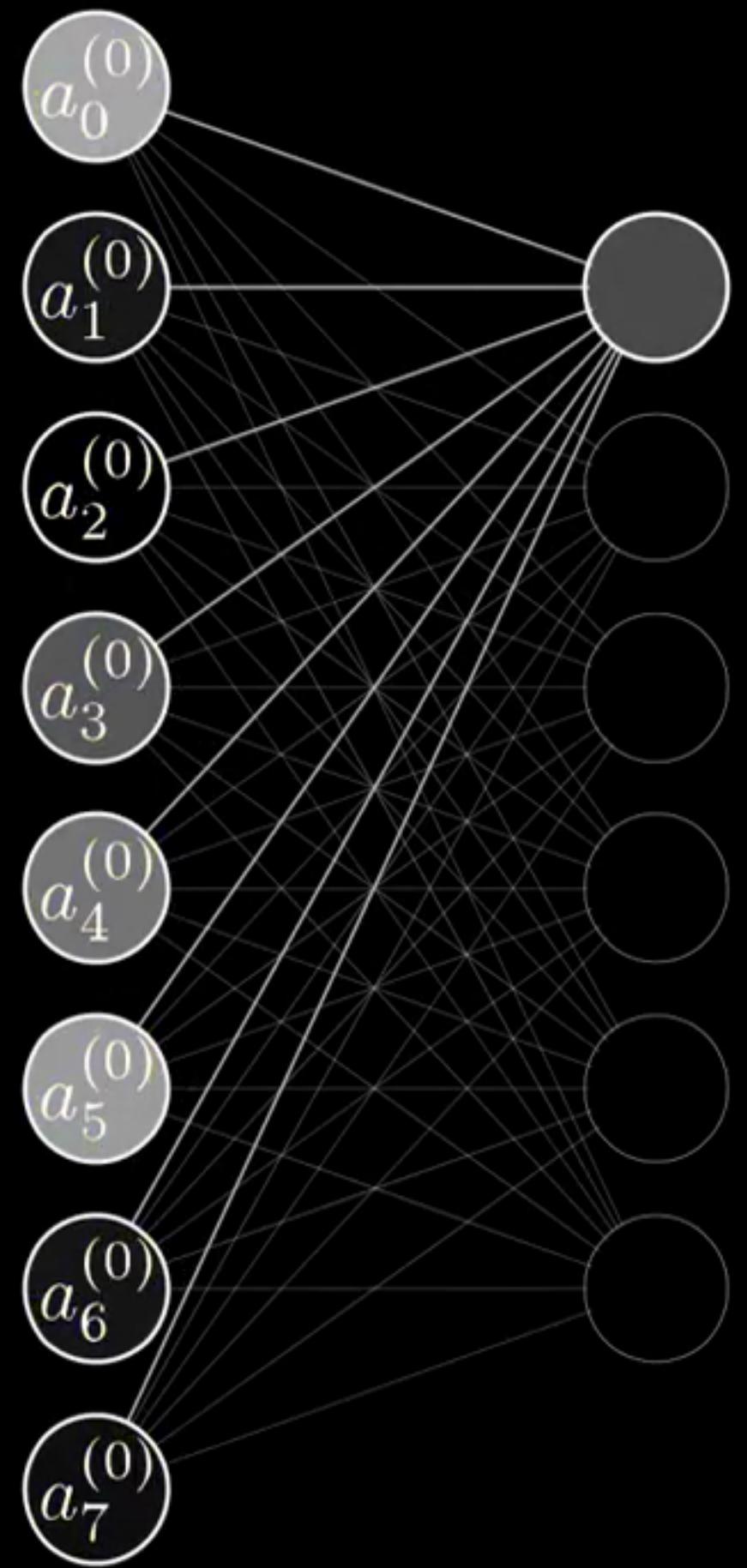
Wrong!



Sigmoid

$$a_0^{(1)} = \sigma(w_{0,0} a_0^{(0)} + w_{0,1} a_1^{(0)} + \cdots + w_{0,n} a_n^{(0)} + b_0)$$

Bias

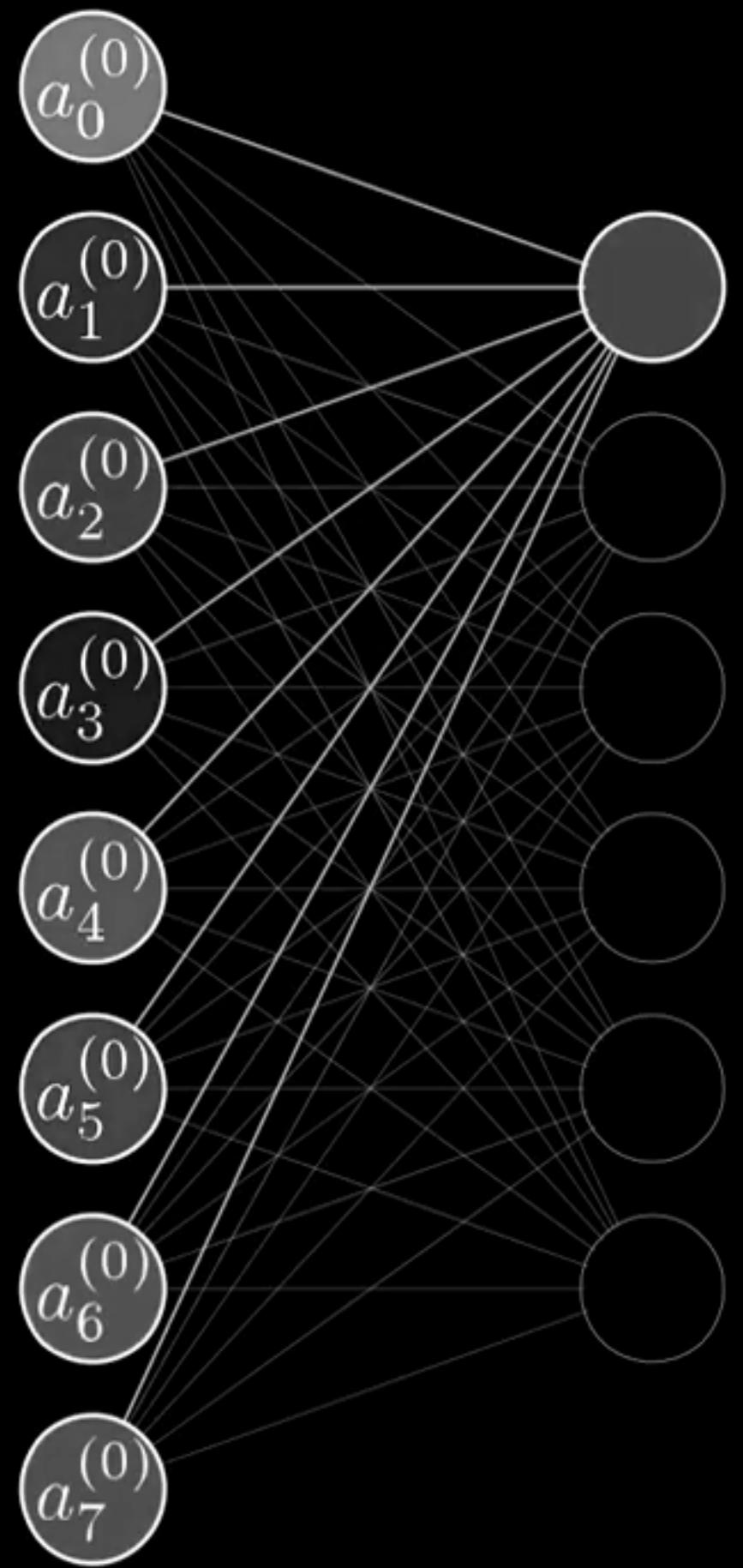


Sigmoid

$$a_0^{(1)} = \sigma(w_{0,0} a_0^{(0)} + w_{0,1} a_1^{(0)} + \cdots + w_{0,n} a_n^{(0)} + b_0)$$

Bias

$$\begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix}$$

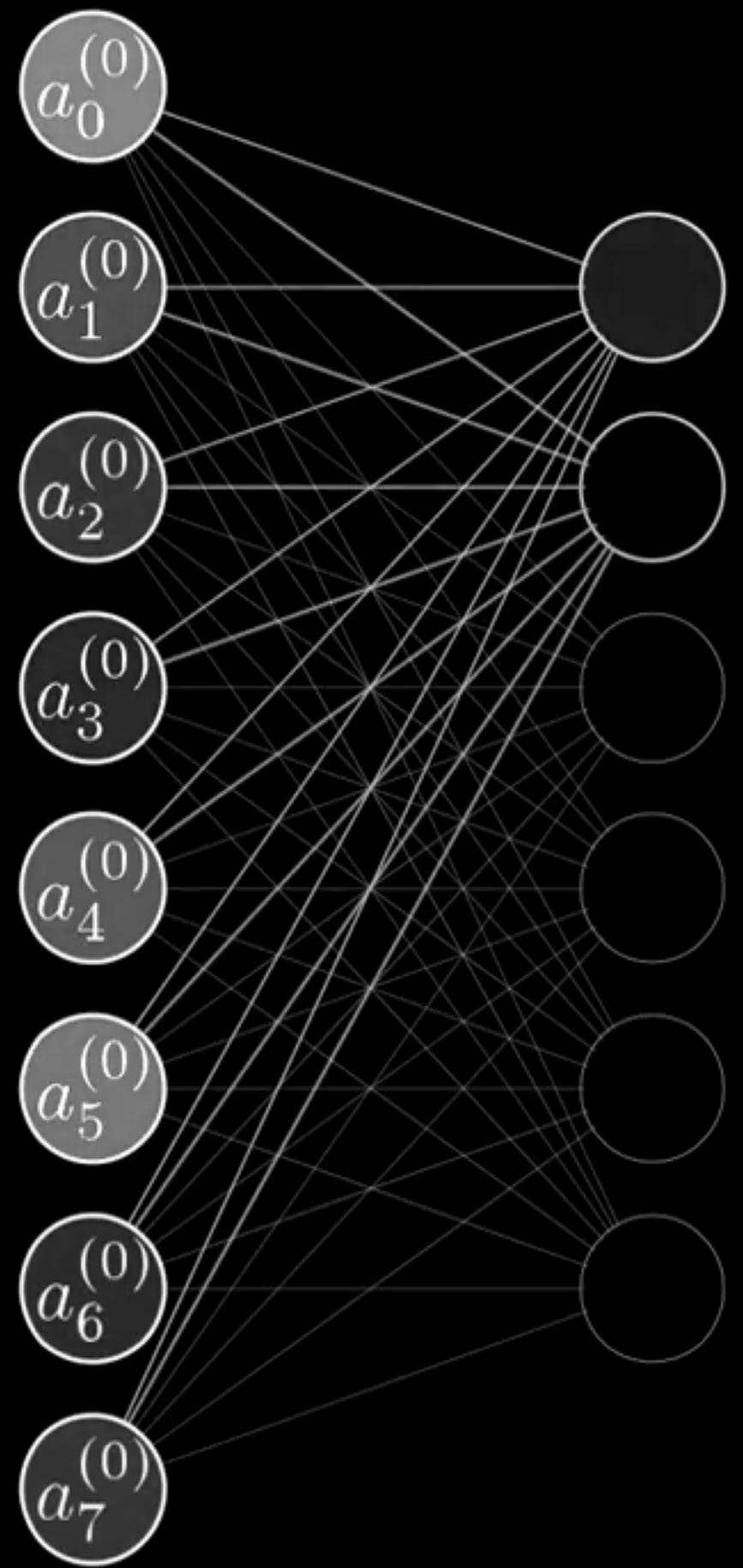


Sigmoid

$$a_0^{(1)} = \sigma \left(w_{0,0} a_0^{(0)} + w_{0,1} a_1^{(0)} + \cdots + w_{0,n} a_n^{(0)} + b_0 \right)$$

Bias

$$\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \dots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ \vdots \\ ? \end{bmatrix}$$

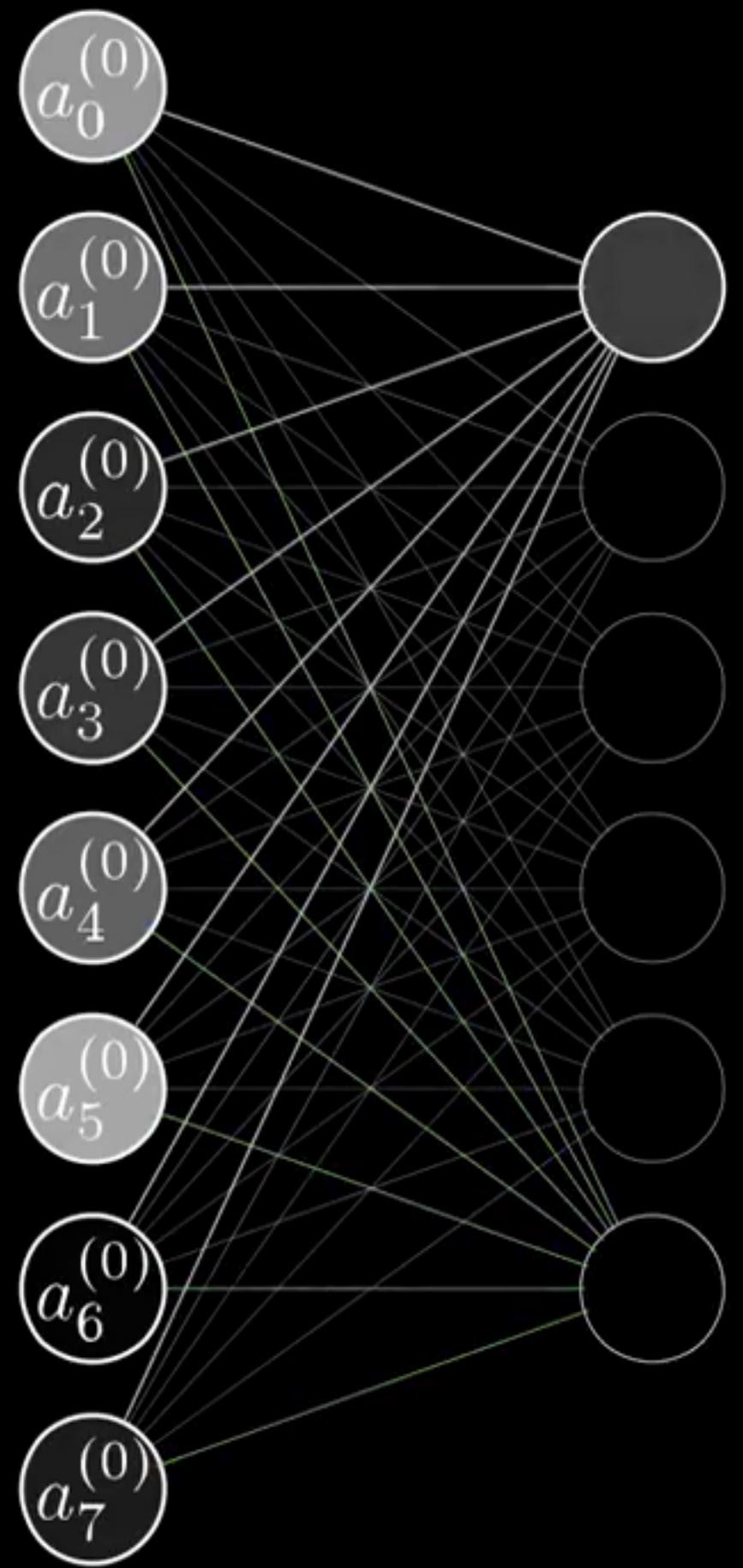


Sigmoid

$$a_0^{(1)} = \sigma \left(w_{0,0} a_0^{(0)} + w_{0,1} a_1^{(0)} + \cdots + w_{0,n} a_n^{(0)} + b_0 \right)$$

Bias

$$\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \dots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ \vdots \\ ? \end{bmatrix}$$

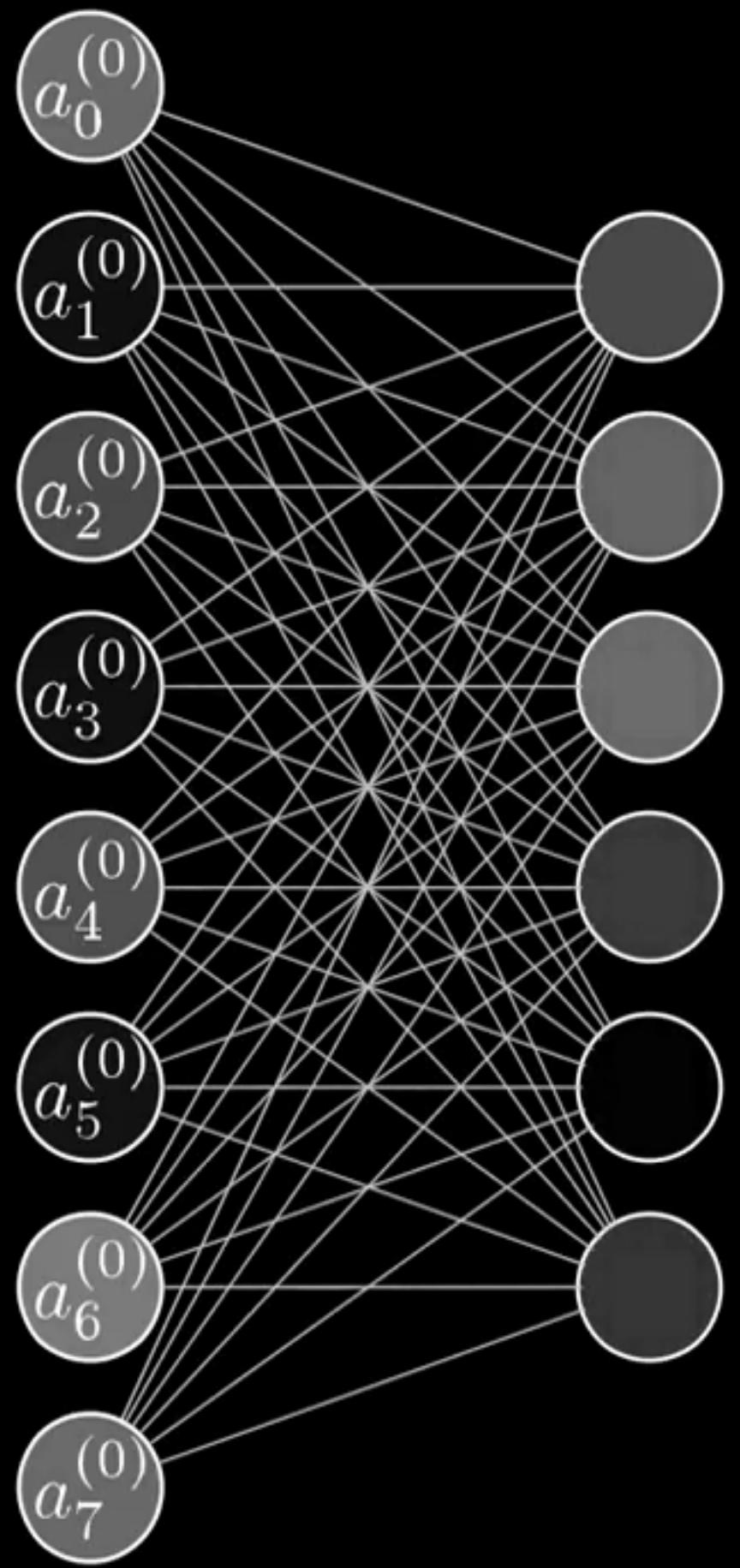


Sigmoid

$$a_0^{(1)} = \sigma(w_{0,0} a_0^{(0)} + w_{0,1} a_1^{(0)} + \cdots + w_{0,n} a_n^{(0)} + b_0)$$

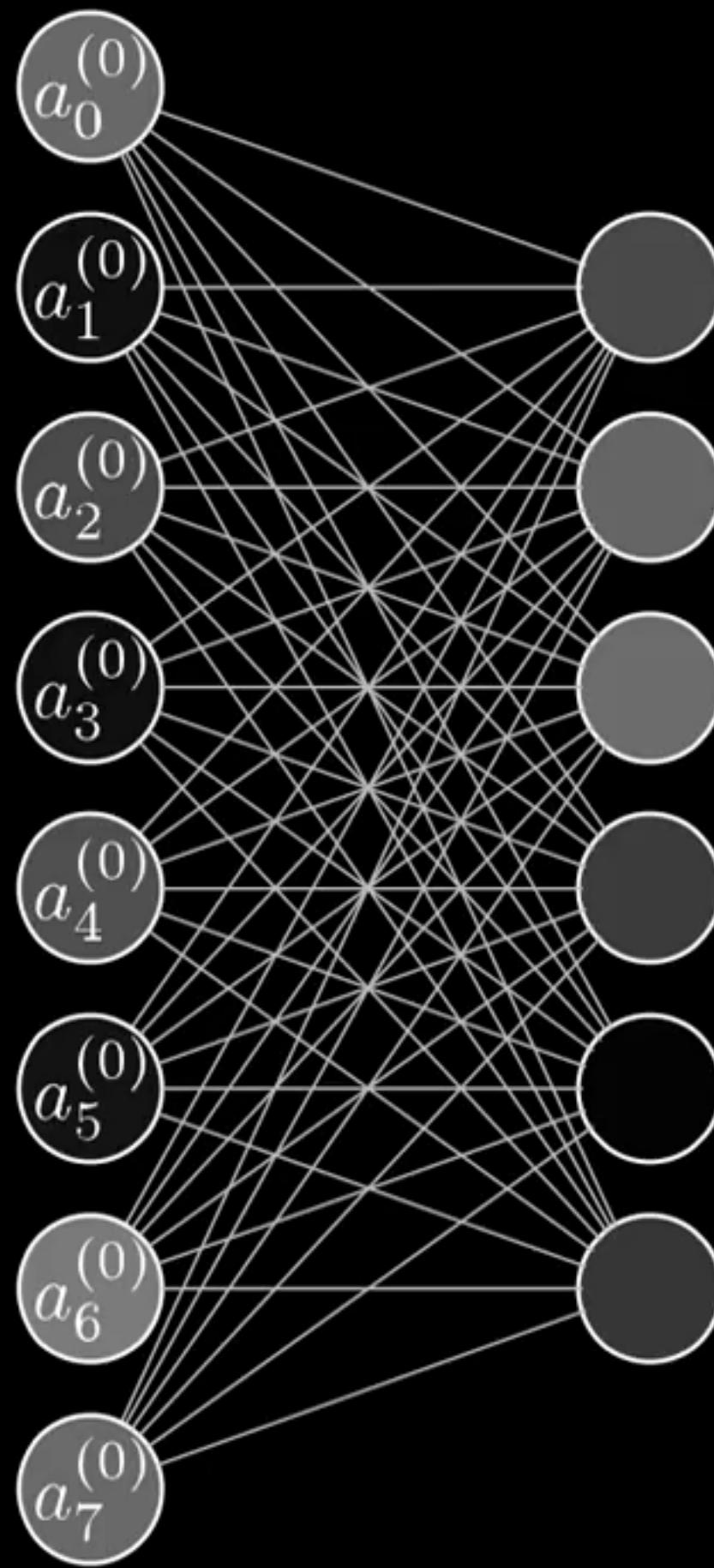
Bias

$$\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \dots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ \vdots \\ ? \end{bmatrix}$$



$$\sigma(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b})$$

$$\sigma \left(\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \dots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \right)$$



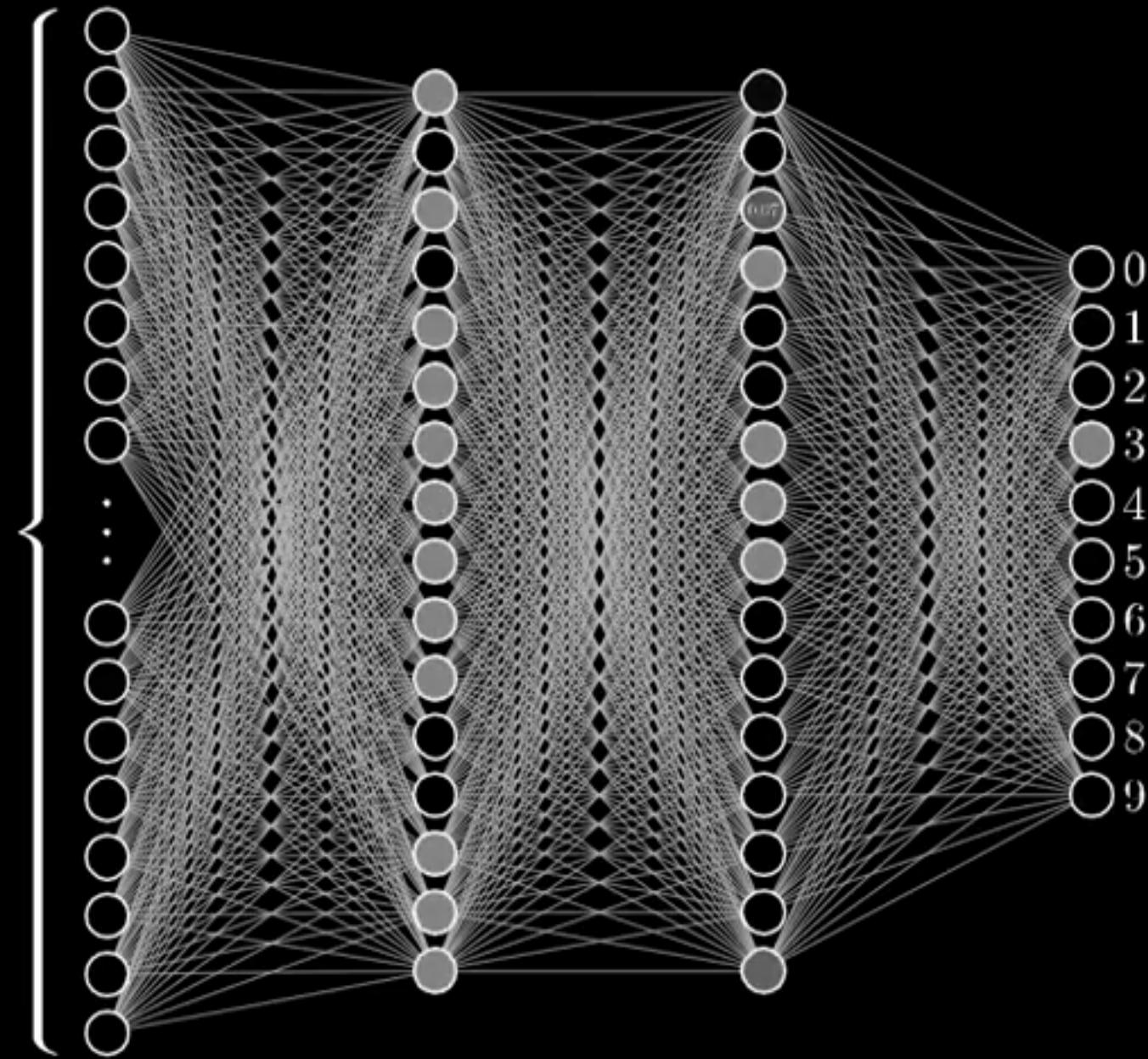
$$\mathbf{a}^{(1)} = \sigma(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b})$$

```
class Network(object):
    def __init__(self, *args, **kwargs):
        #...yada yada, initialize weights and biases...

    def feedforward(self, a):
        """Return the output of the network for an input vector a"""
        for b, w in zip(self.biases, self.weights):
            a = sigmoid(np.dot(w, a) + b)
        return a
```



784



Neuron



Thing that holds
a number

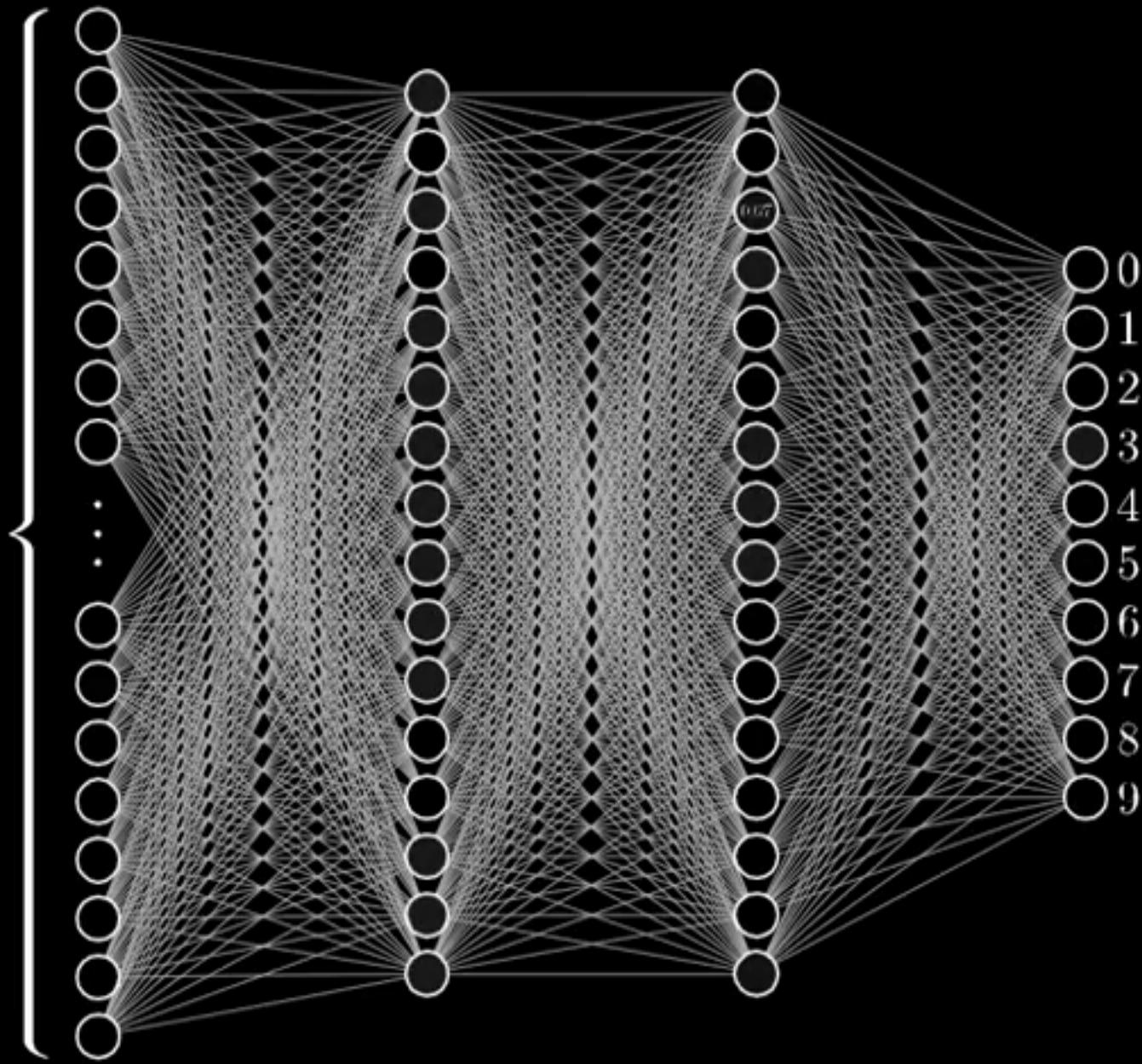
Neuron



Thing that holds
a number



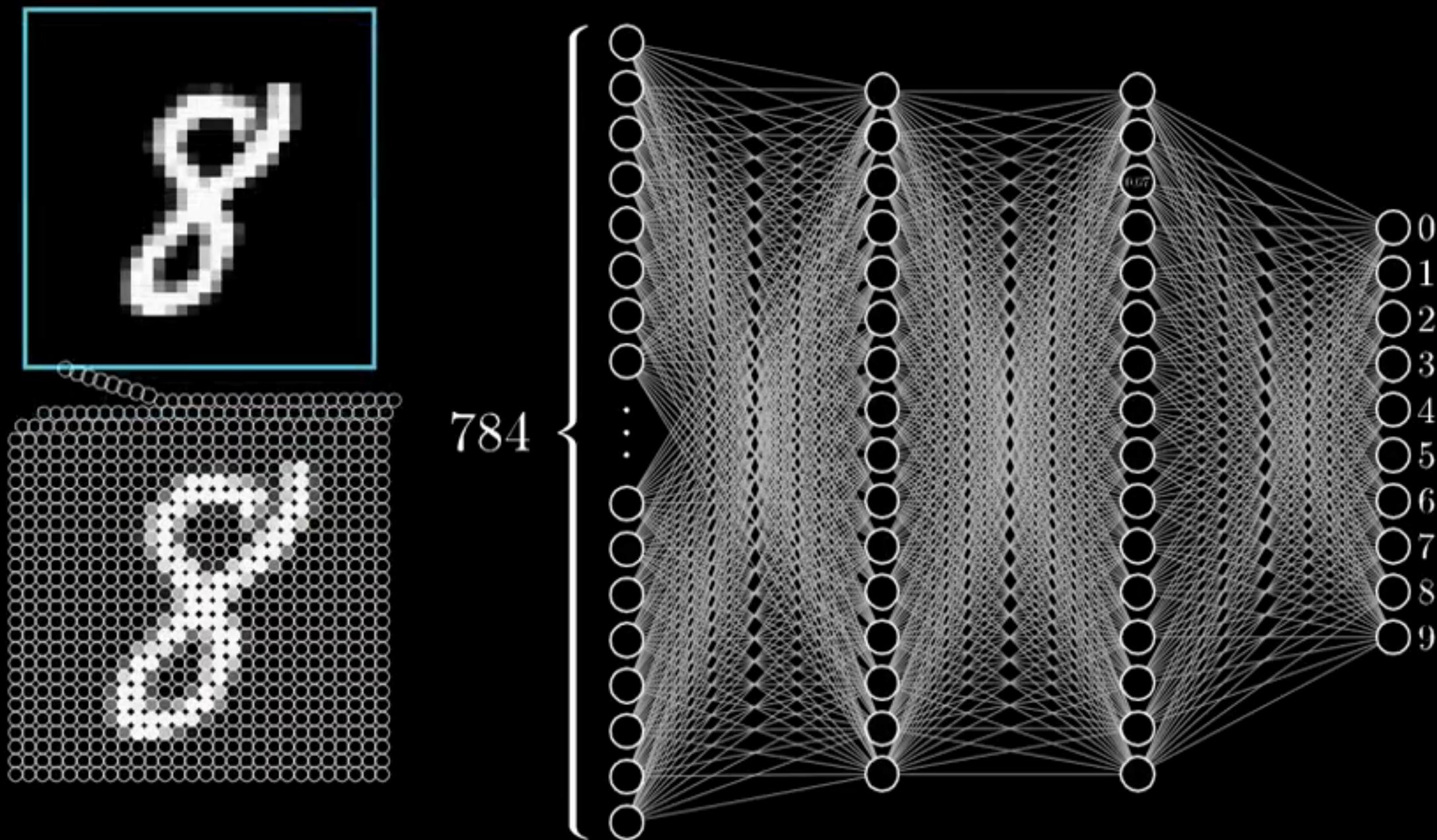
784



Neuron



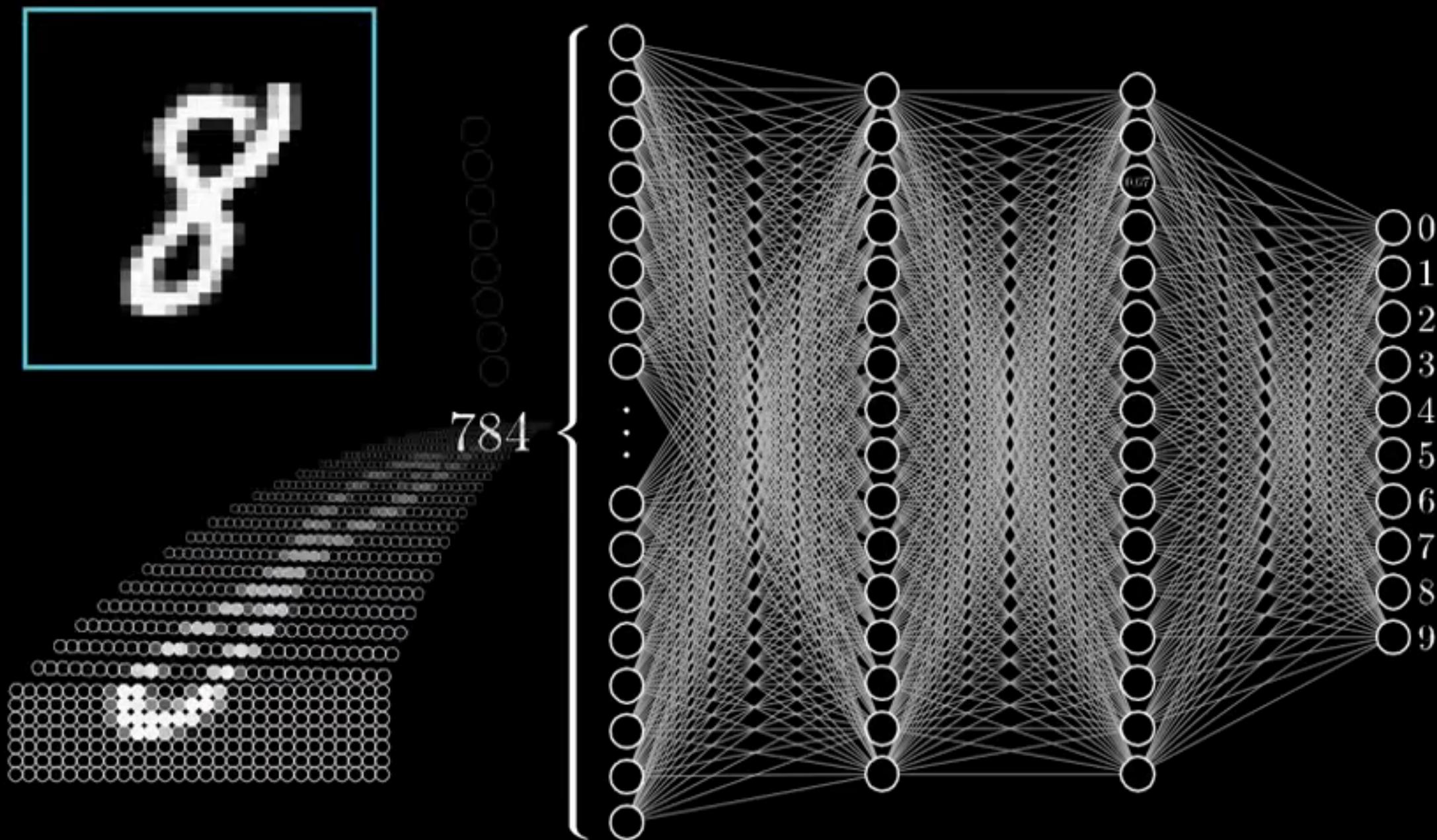
Thing that holds
a number



Neuron



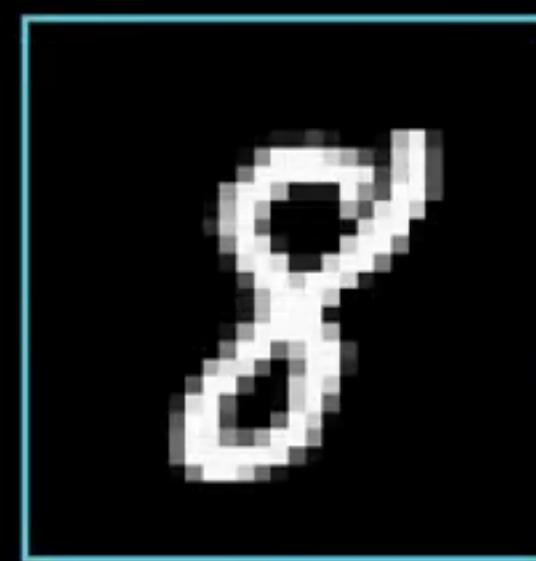
Thing that holds
a number



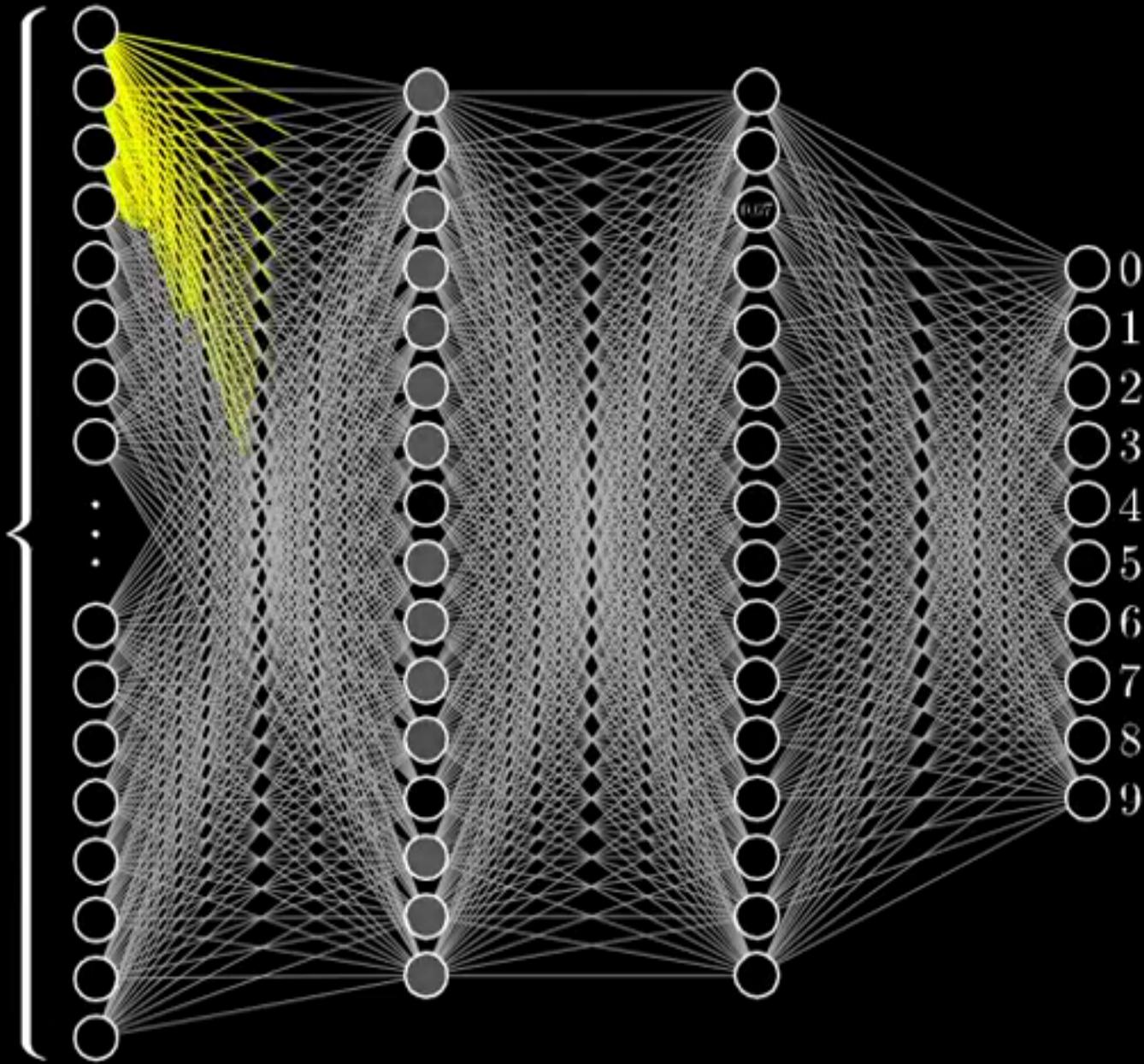
Neuron

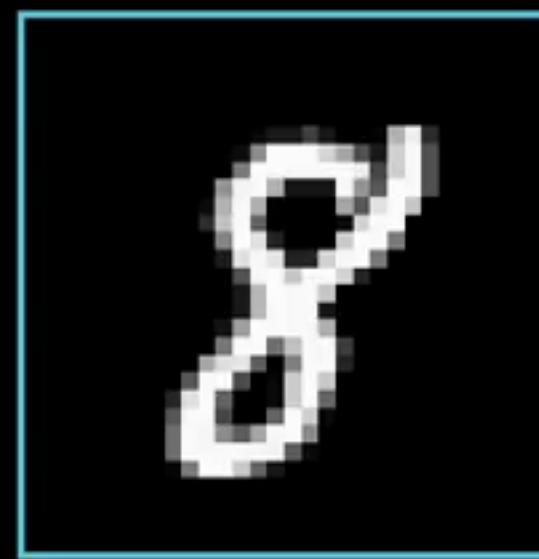


Thing that holds
a number

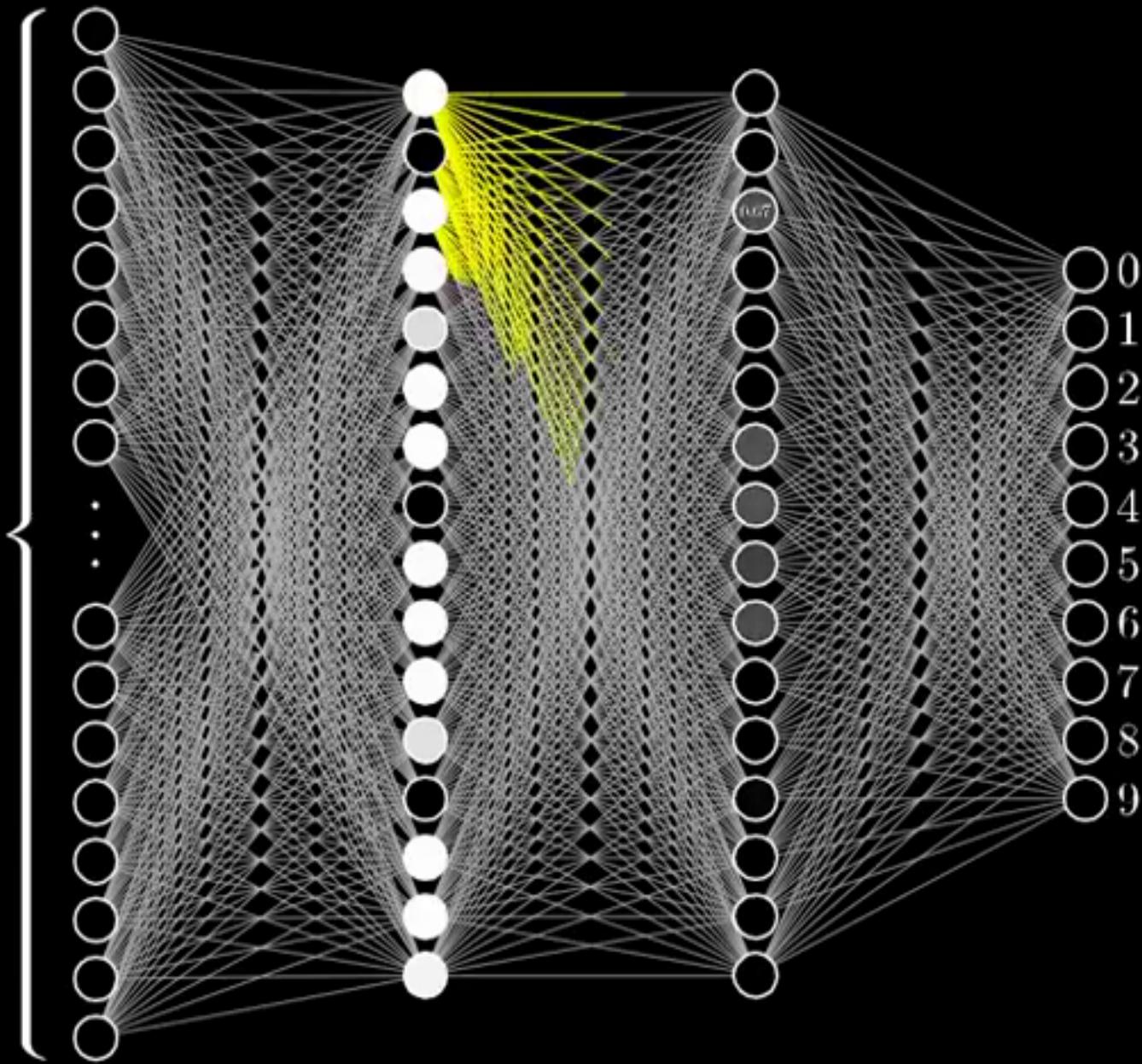


784





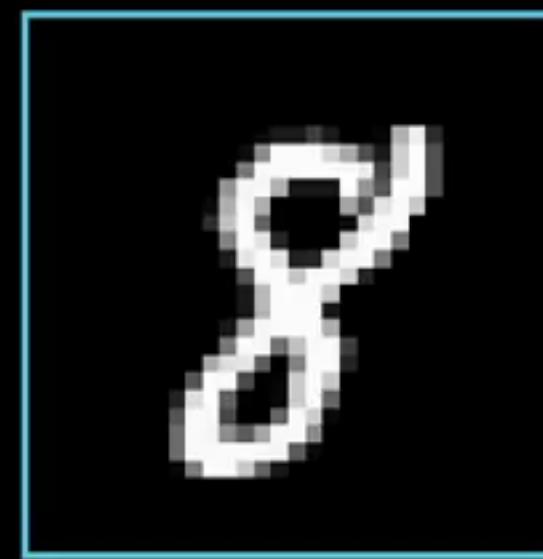
784



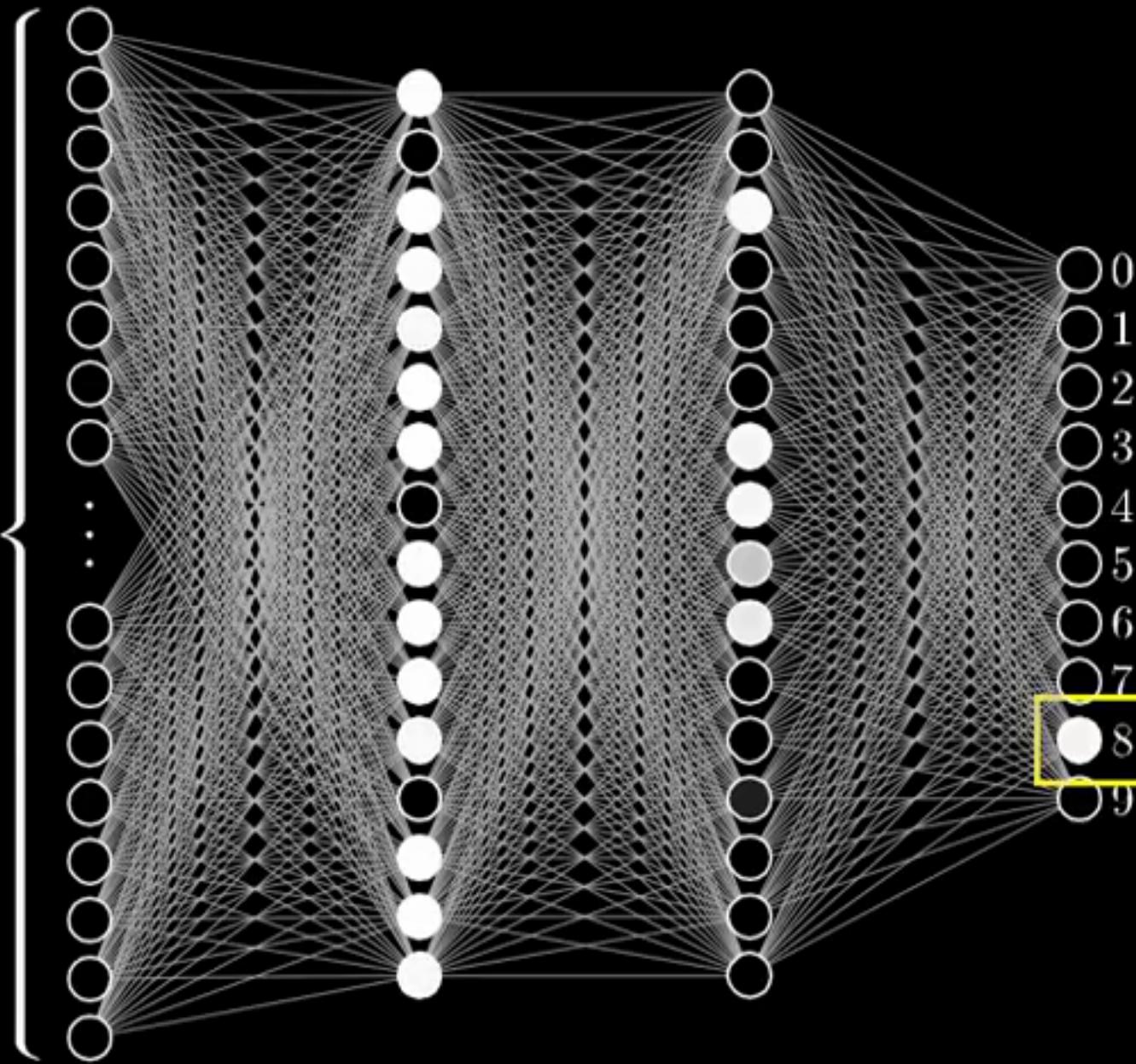
Neuron



Thing that holds
a number



784



Neuron



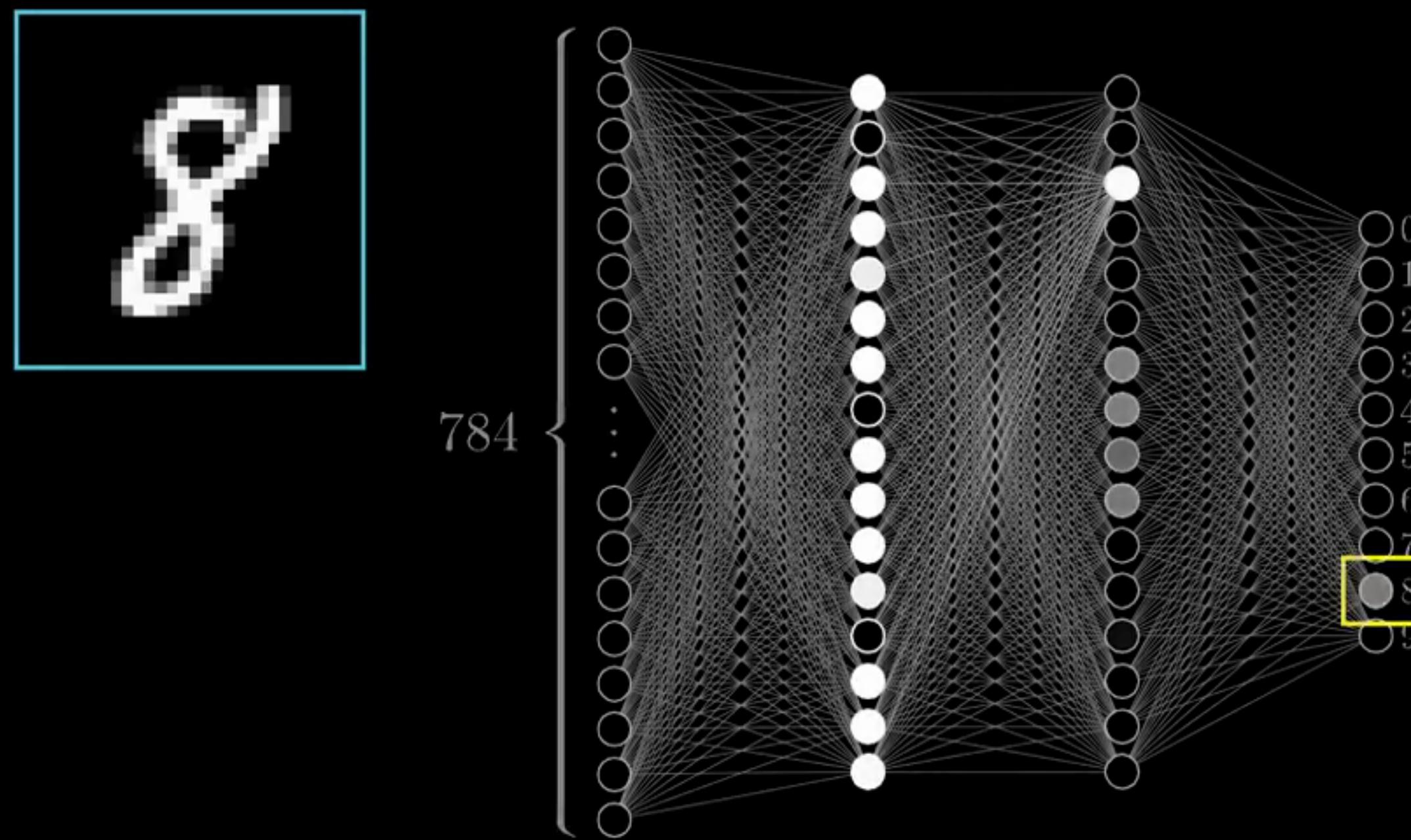
Function

~~Thing that holds
a number~~

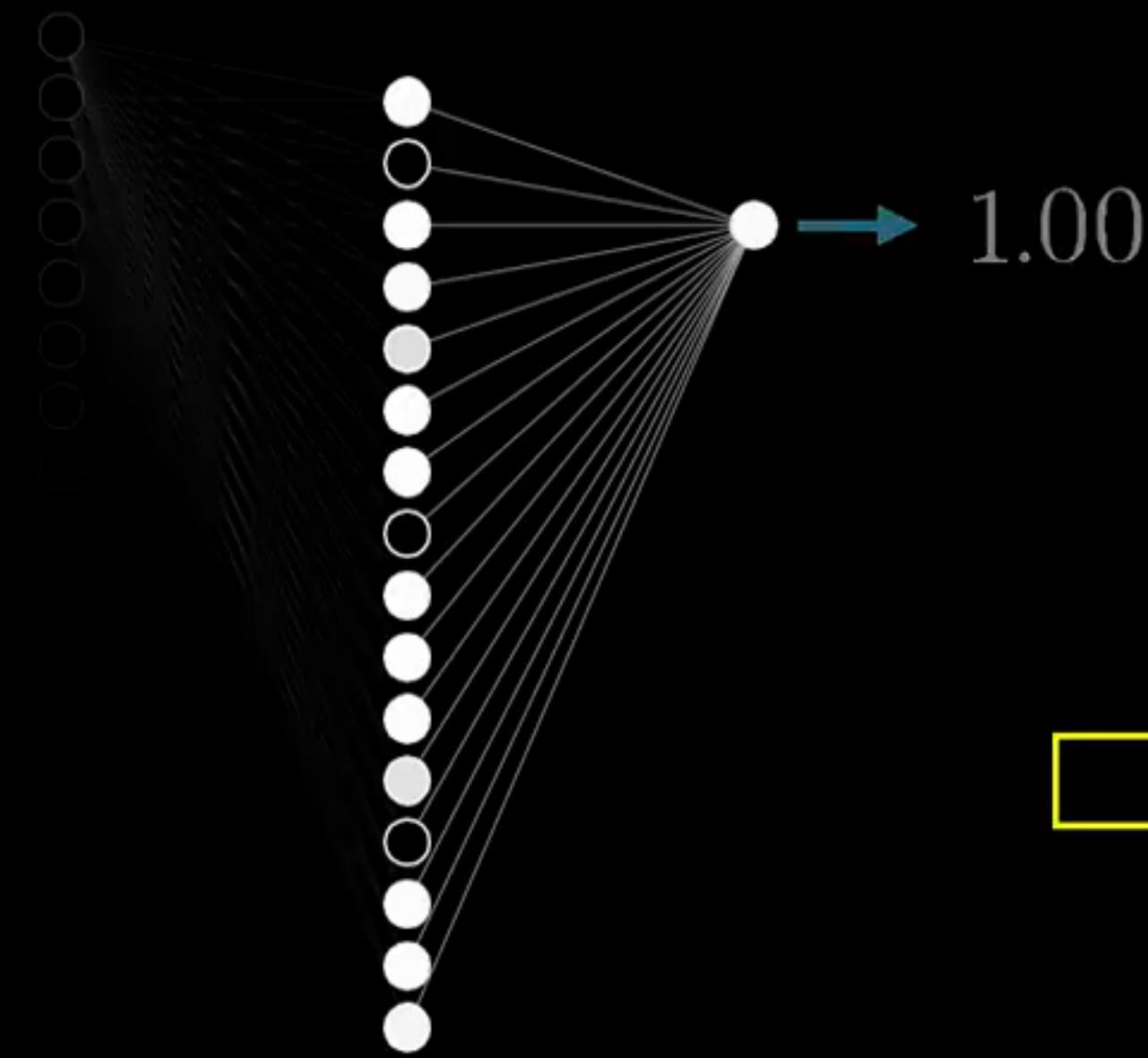
Neuron



Function



~~Thing that holds
a number~~



Neuron



Function

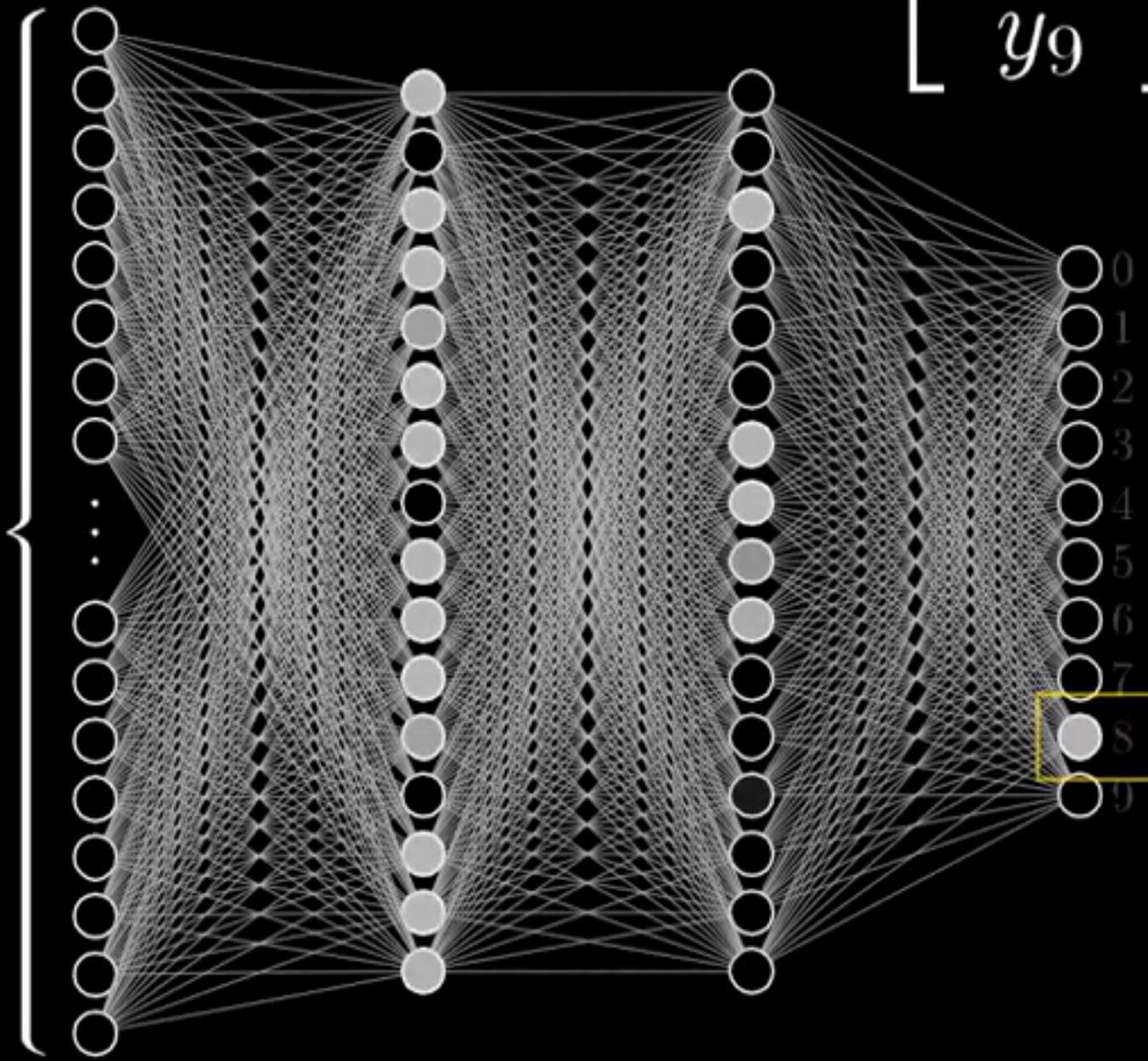


~~Thing that holds
a number~~

$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$

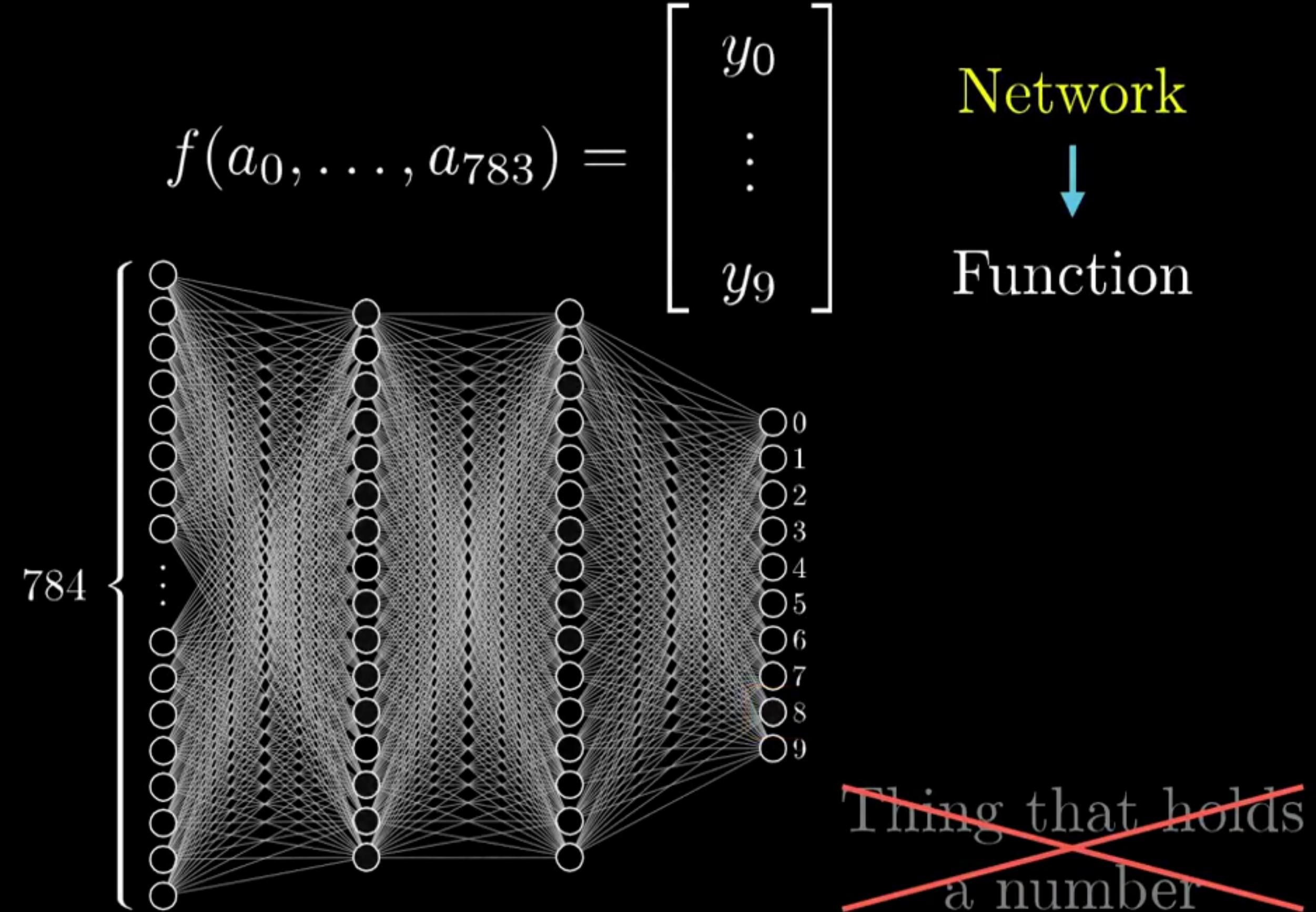
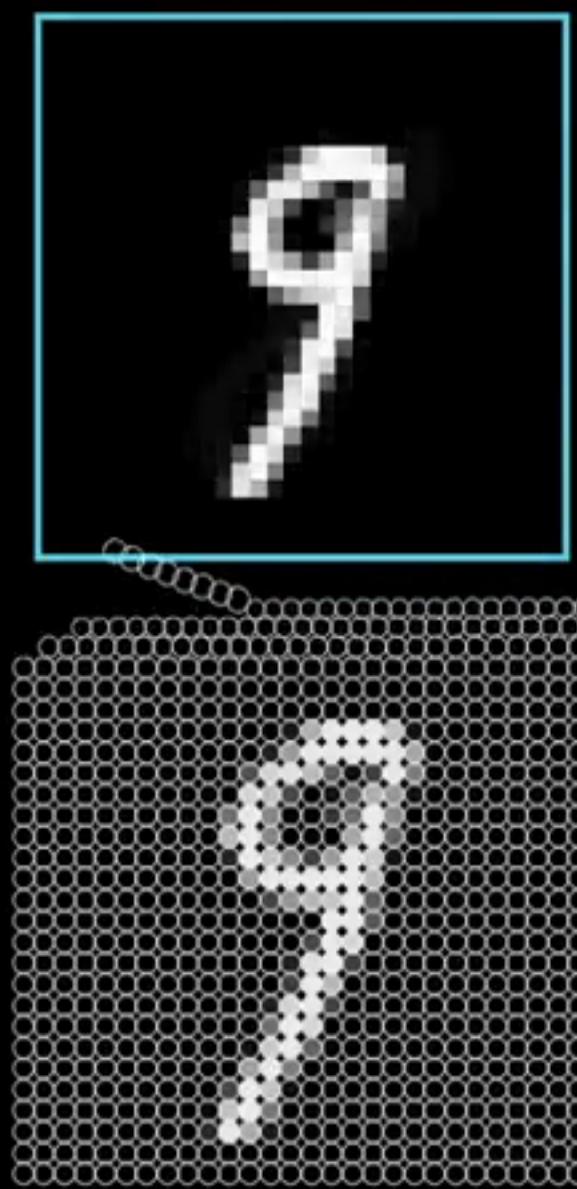


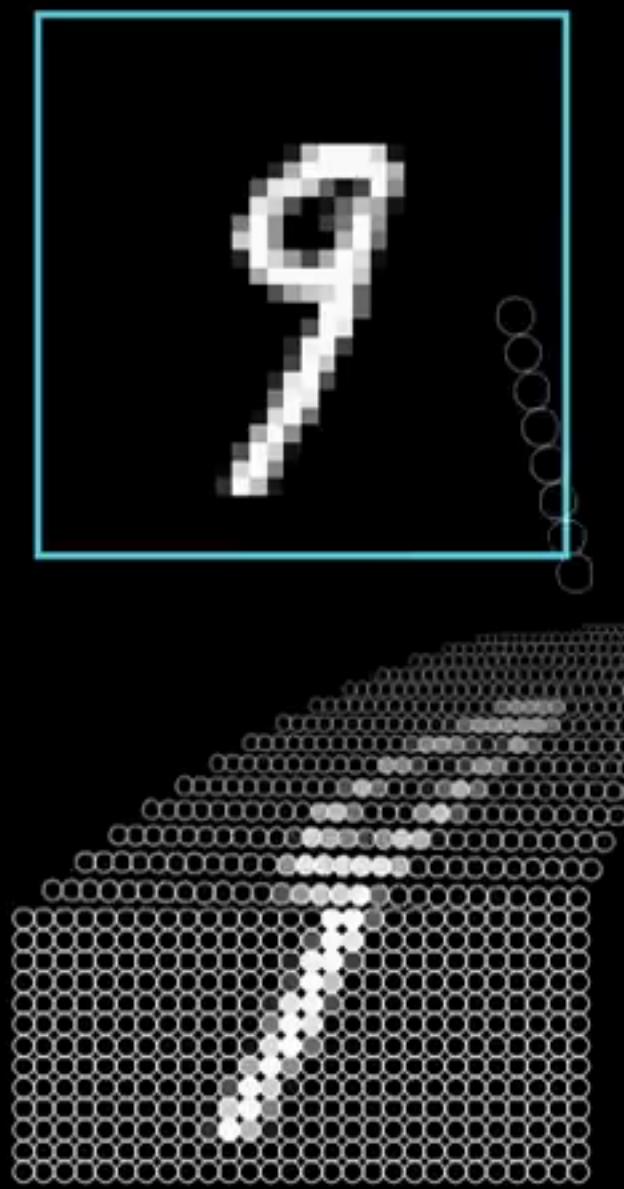
784



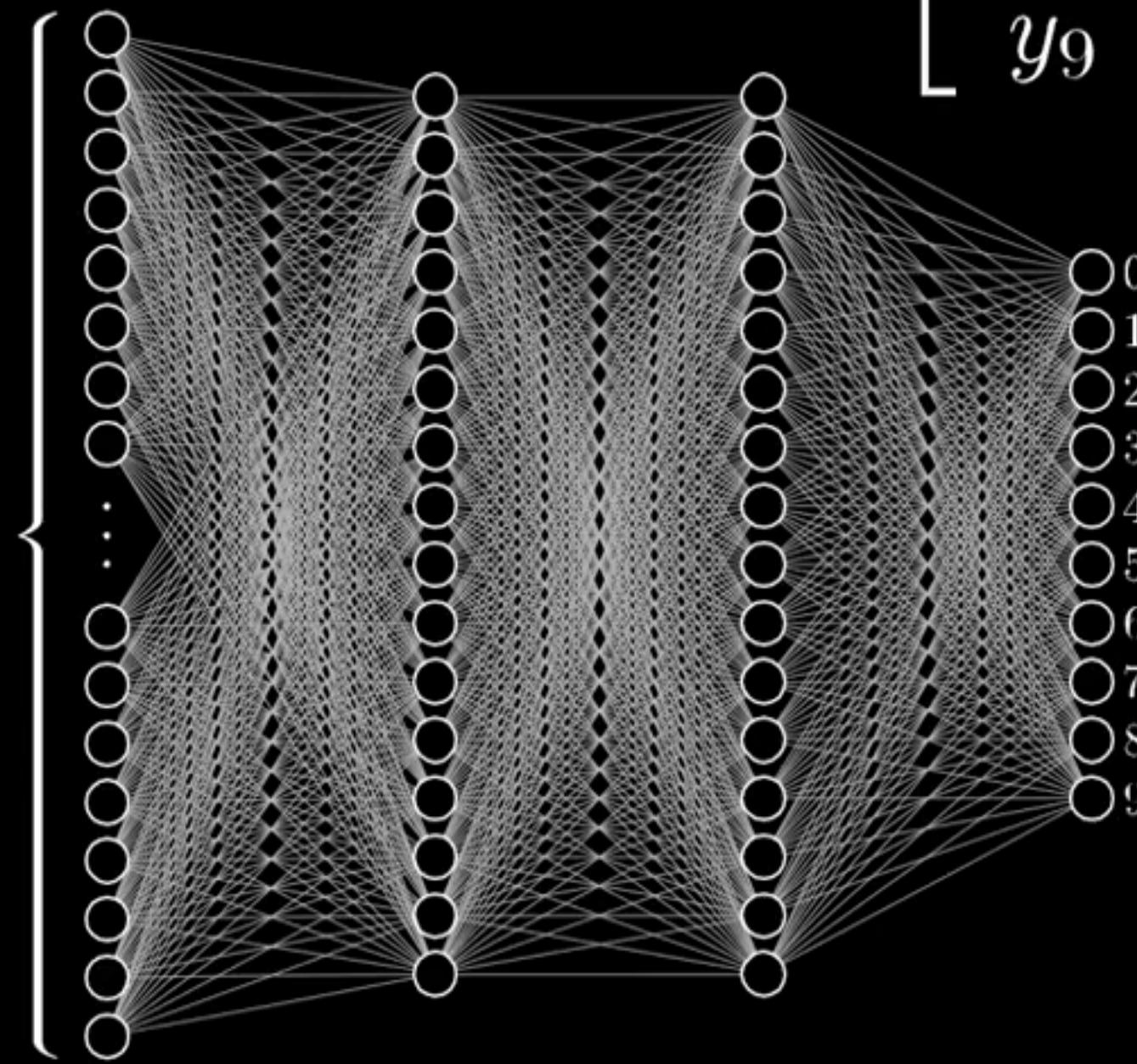
Network
↓
Function

~~Thing that holds
a number~~





784



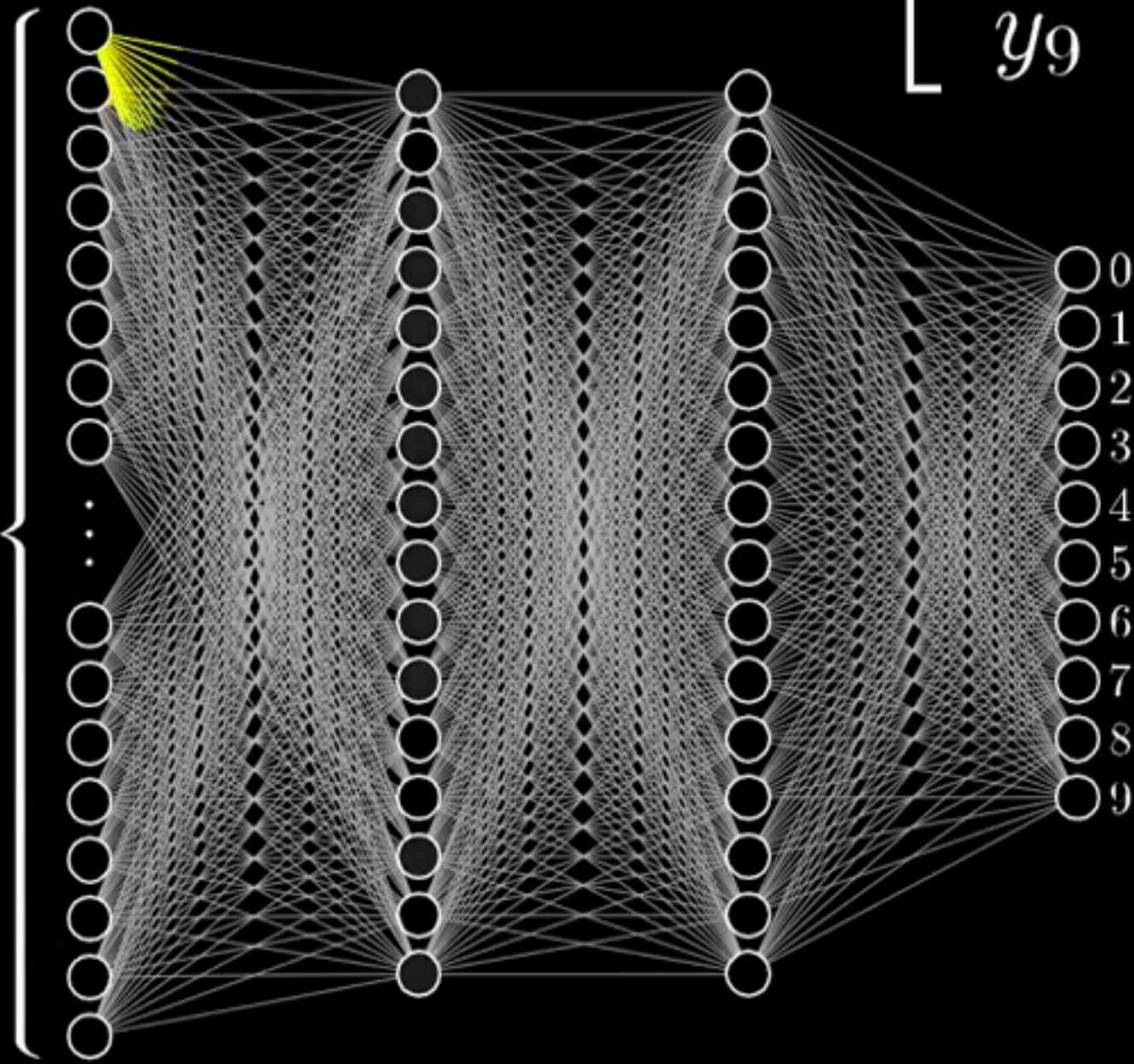
$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$

Network
↓
Function

~~Thing that holds
a number~~



784



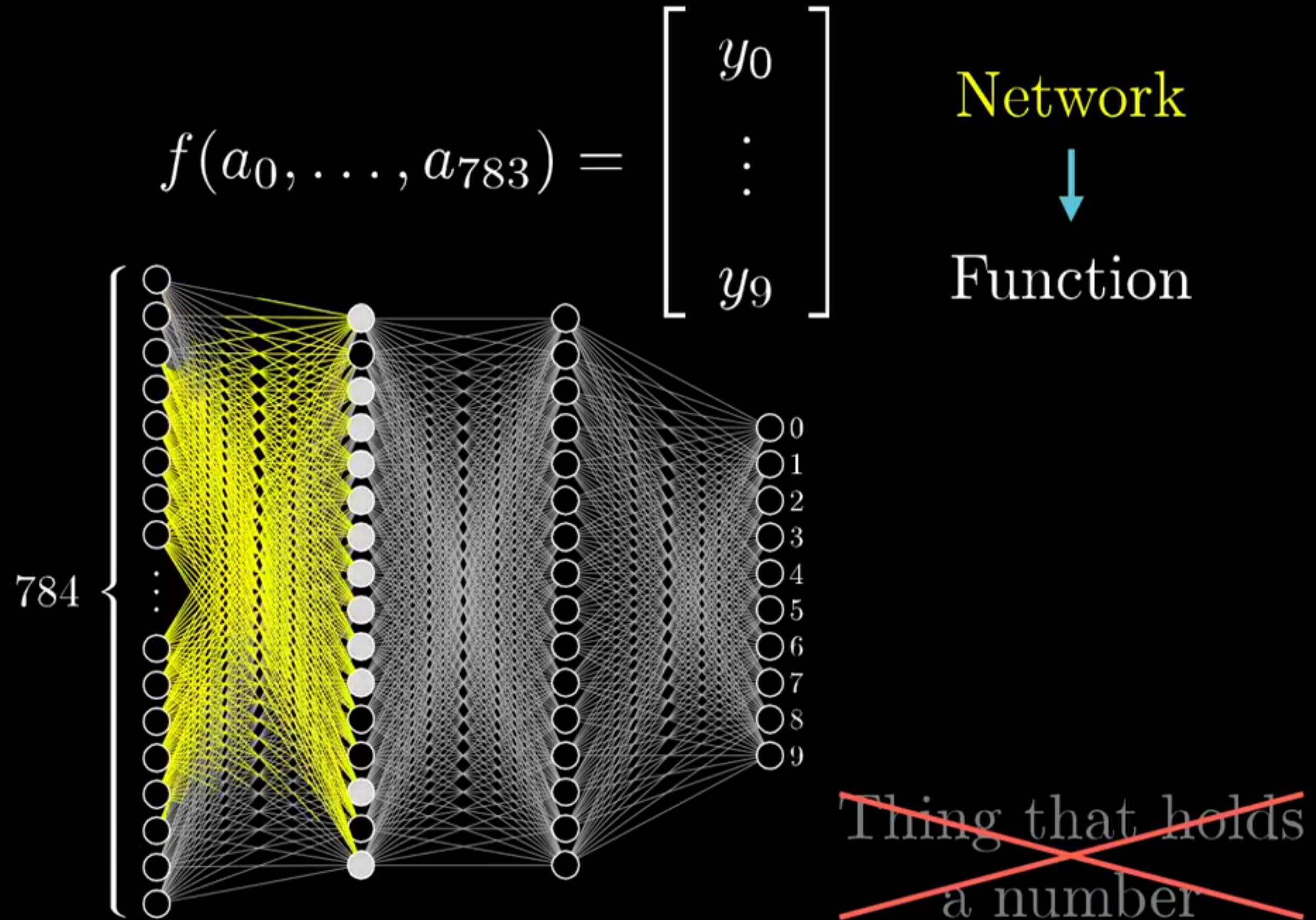
$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$

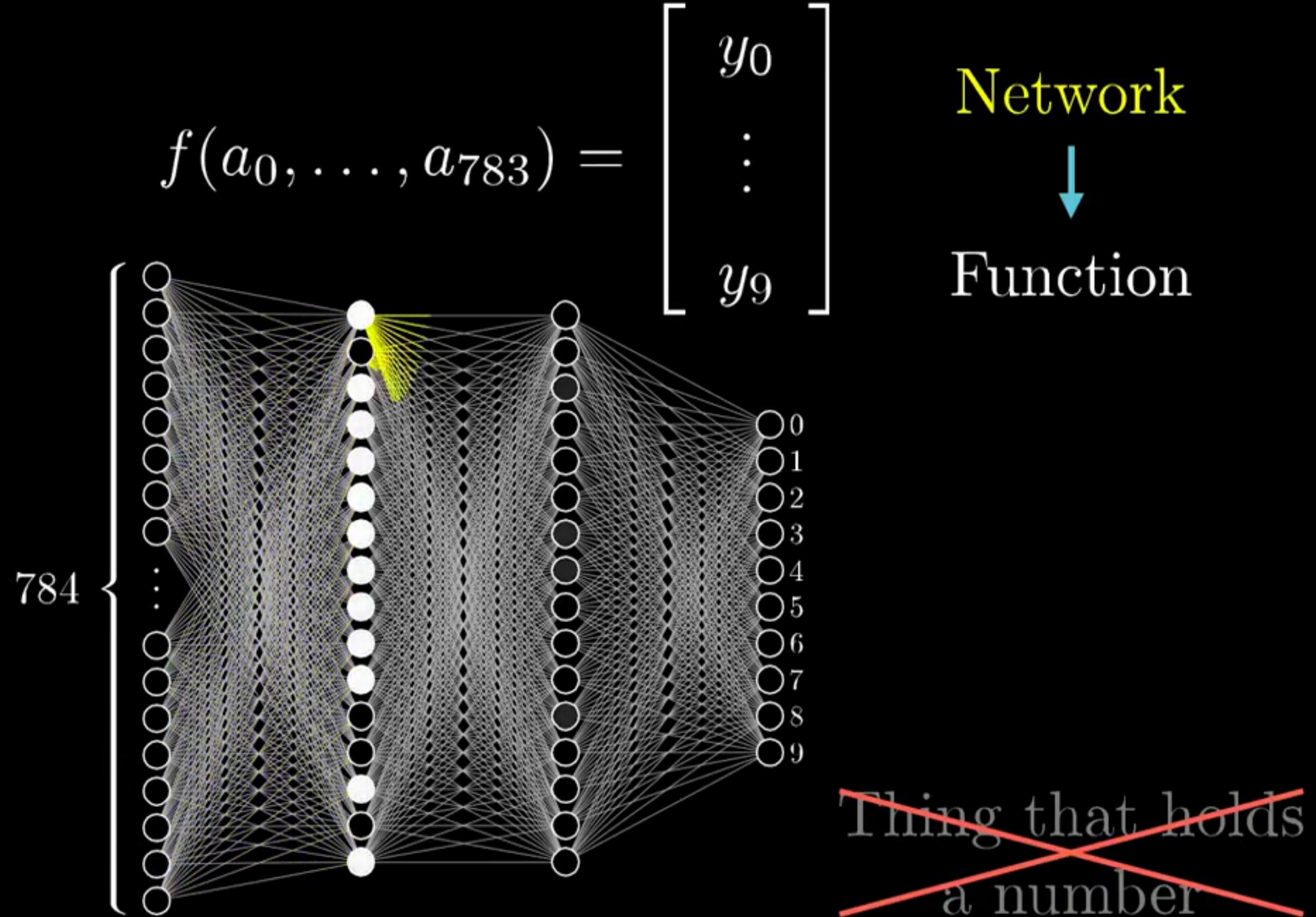
Network

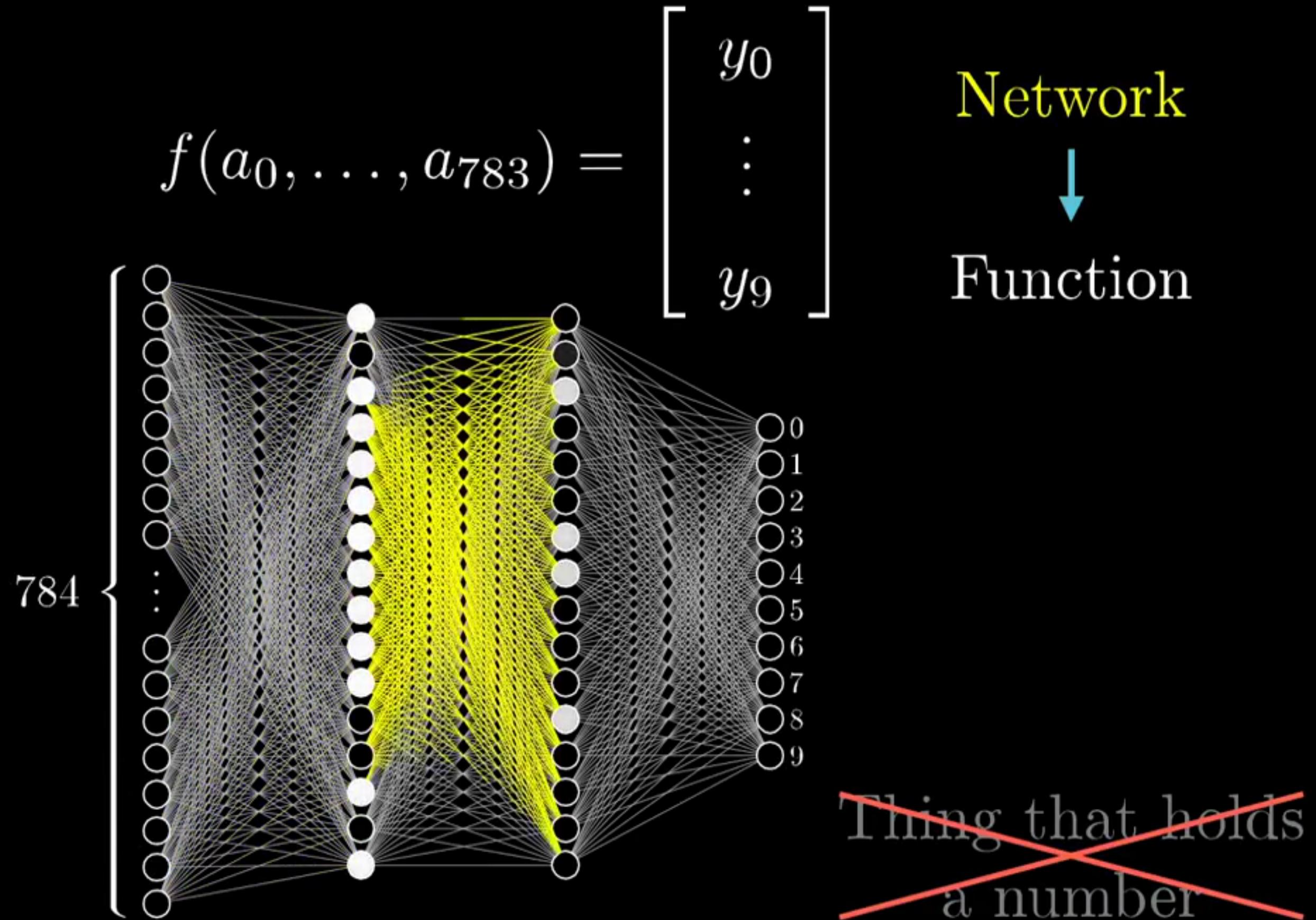


Function

Thing that holds
a number

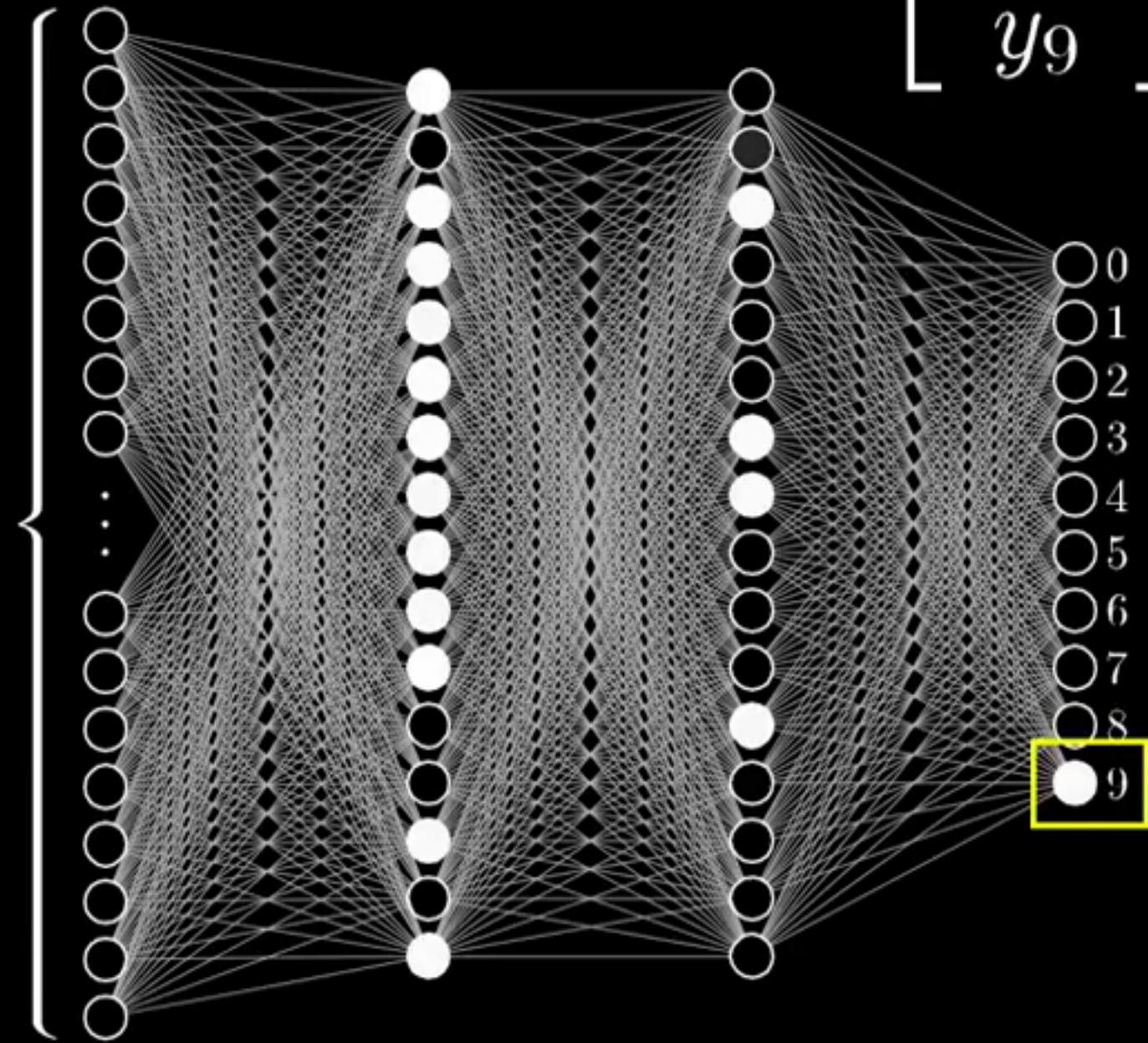






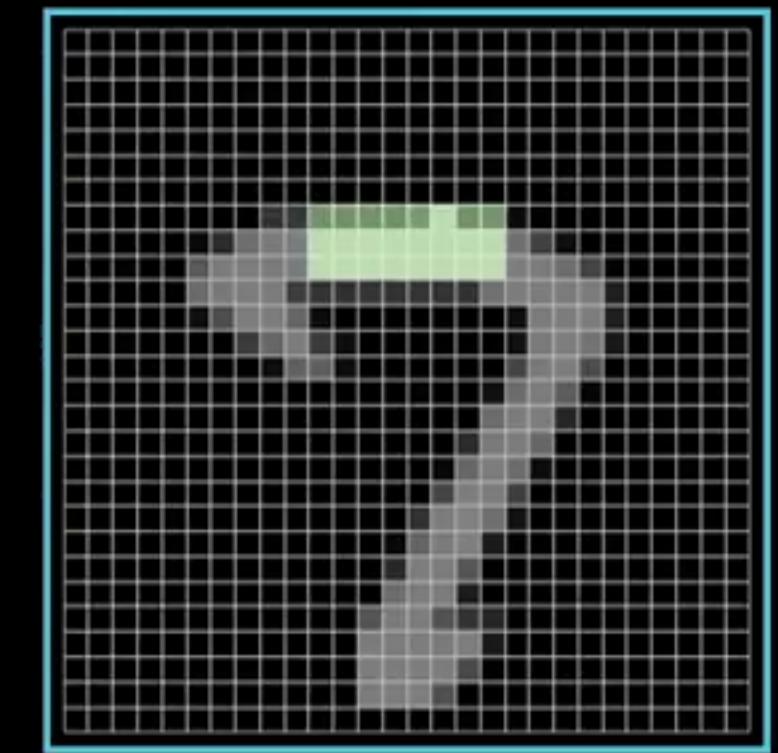
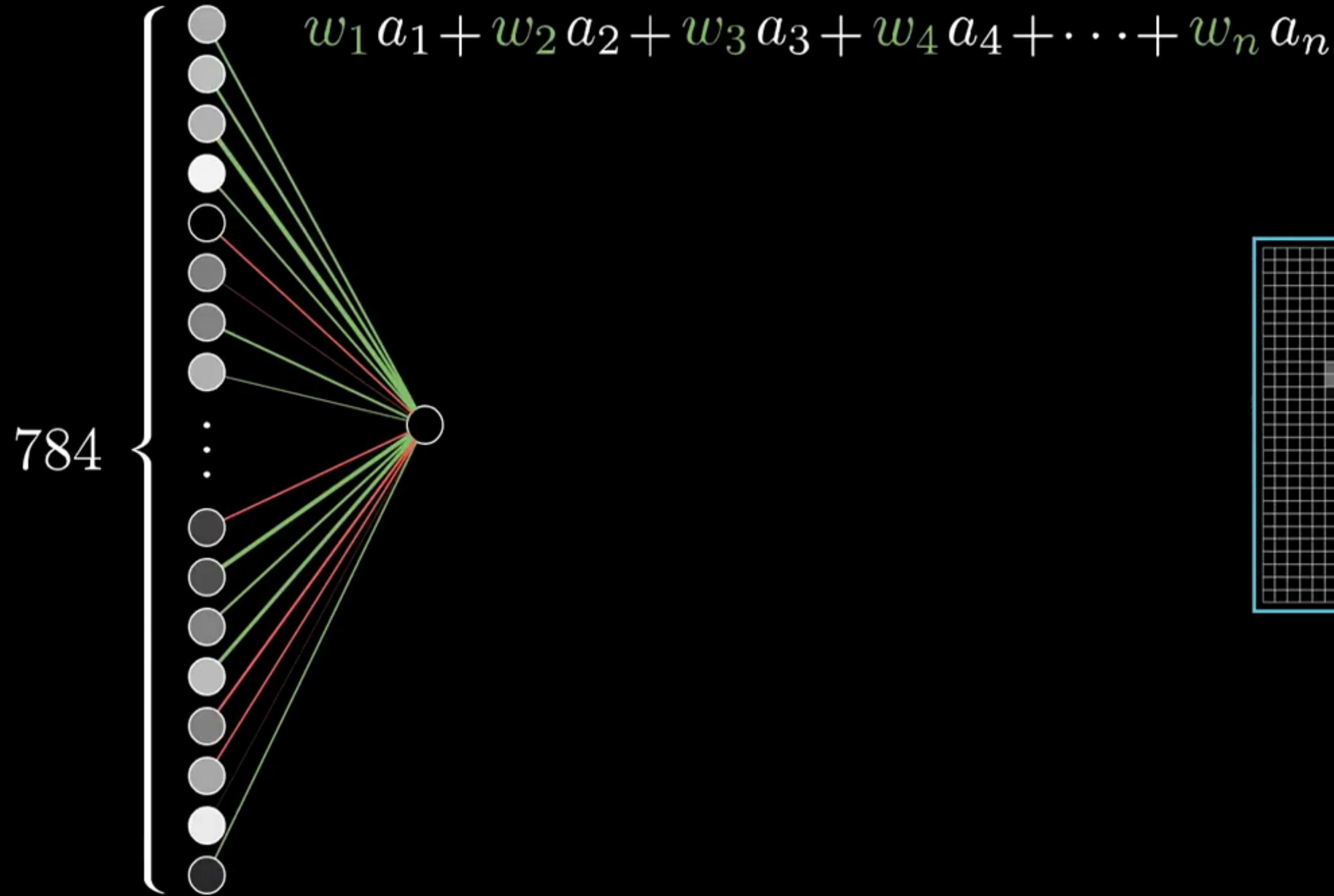


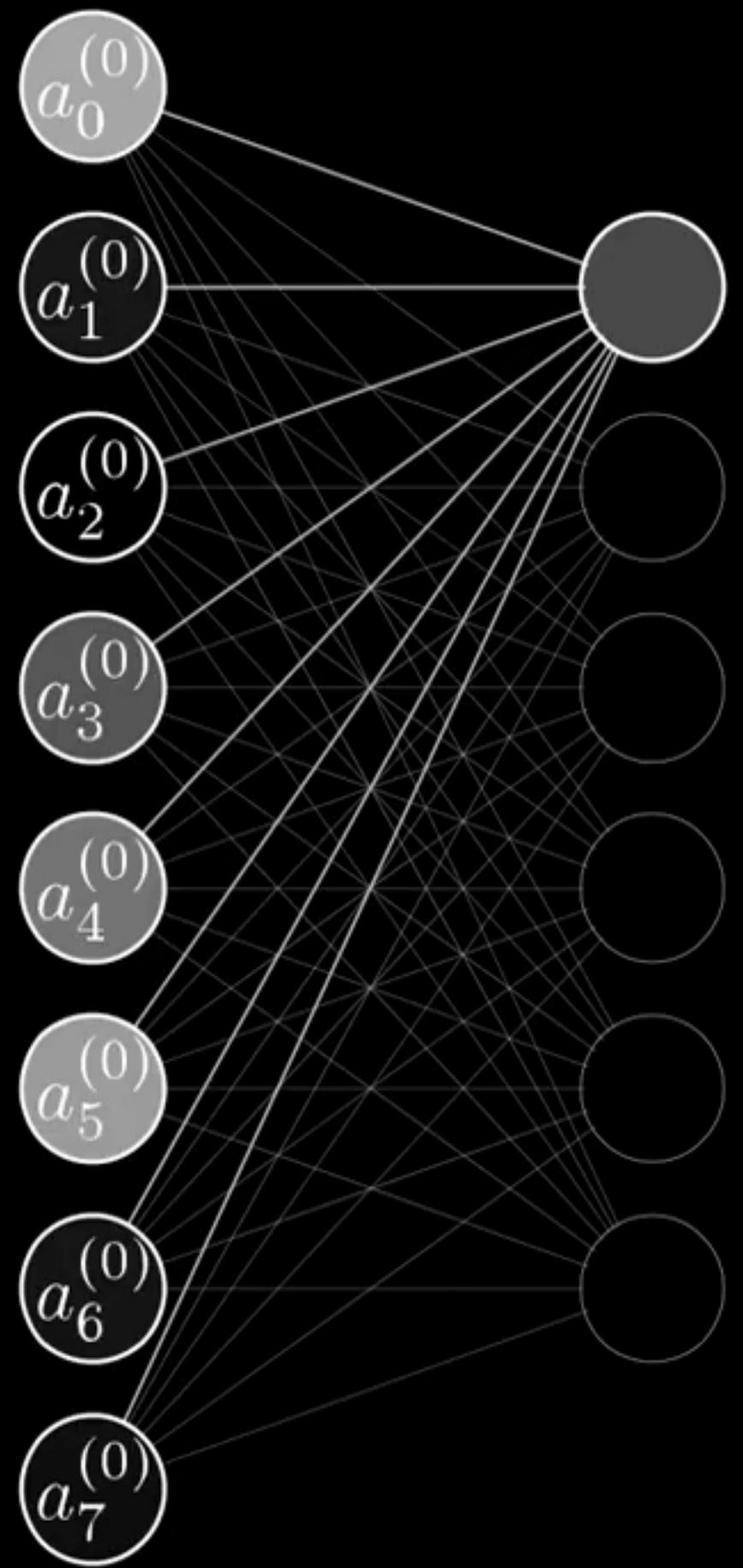
$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$



Network
↓
Function

~~Thing that holds
a number~~





Sigmoid

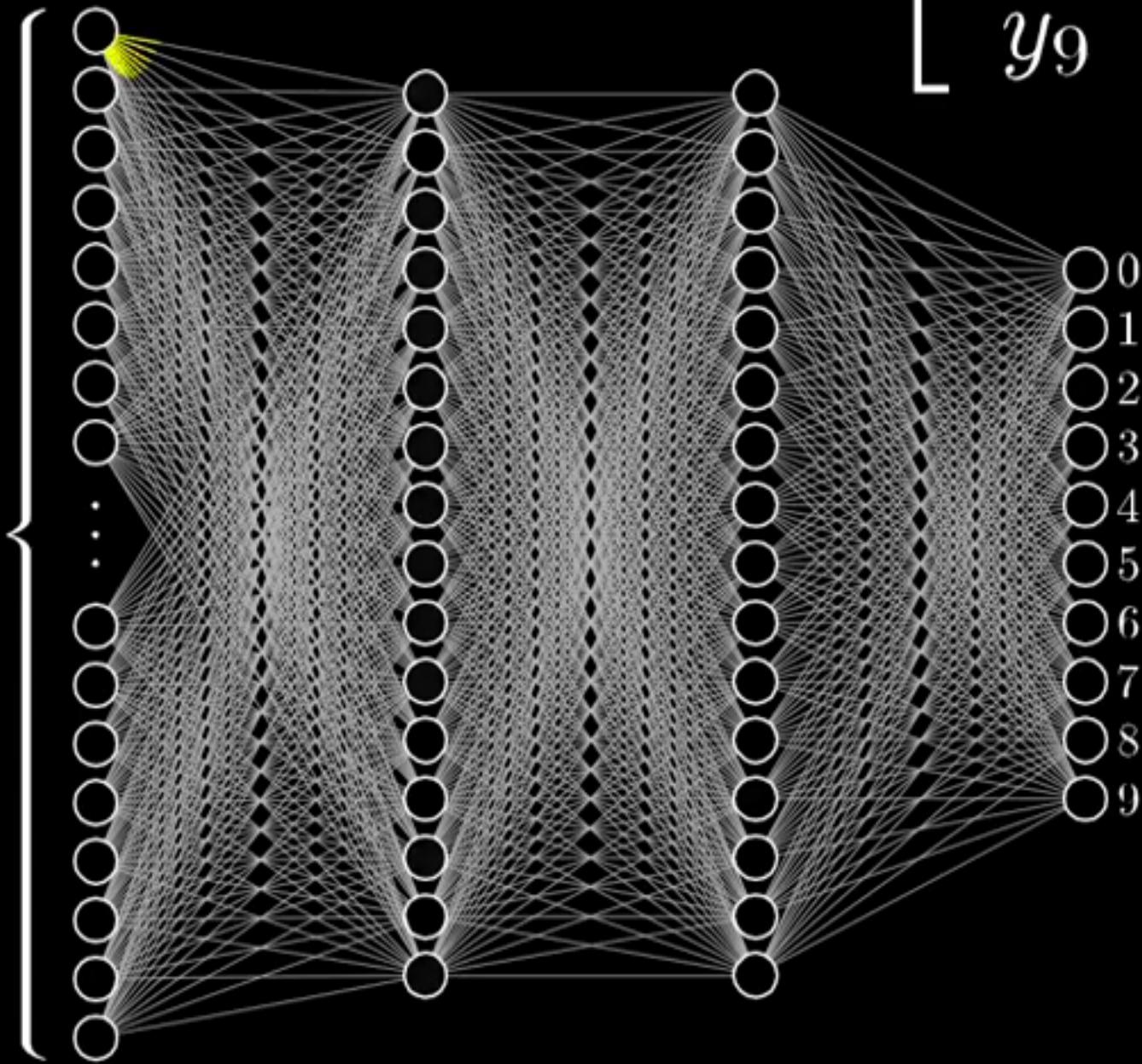
$$a_0^{(1)} = \sigma \left(w_{0,0} a_0^{(0)} + w_{0,1} a_1^{(0)} + \cdots + w_{0,n} a_n^{(0)} + b_0 \right)$$

Bias

$$\sigma \left(\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,0} & w_{n,1} & \dots & w_{n,k} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \right)$$



784



$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$

Network

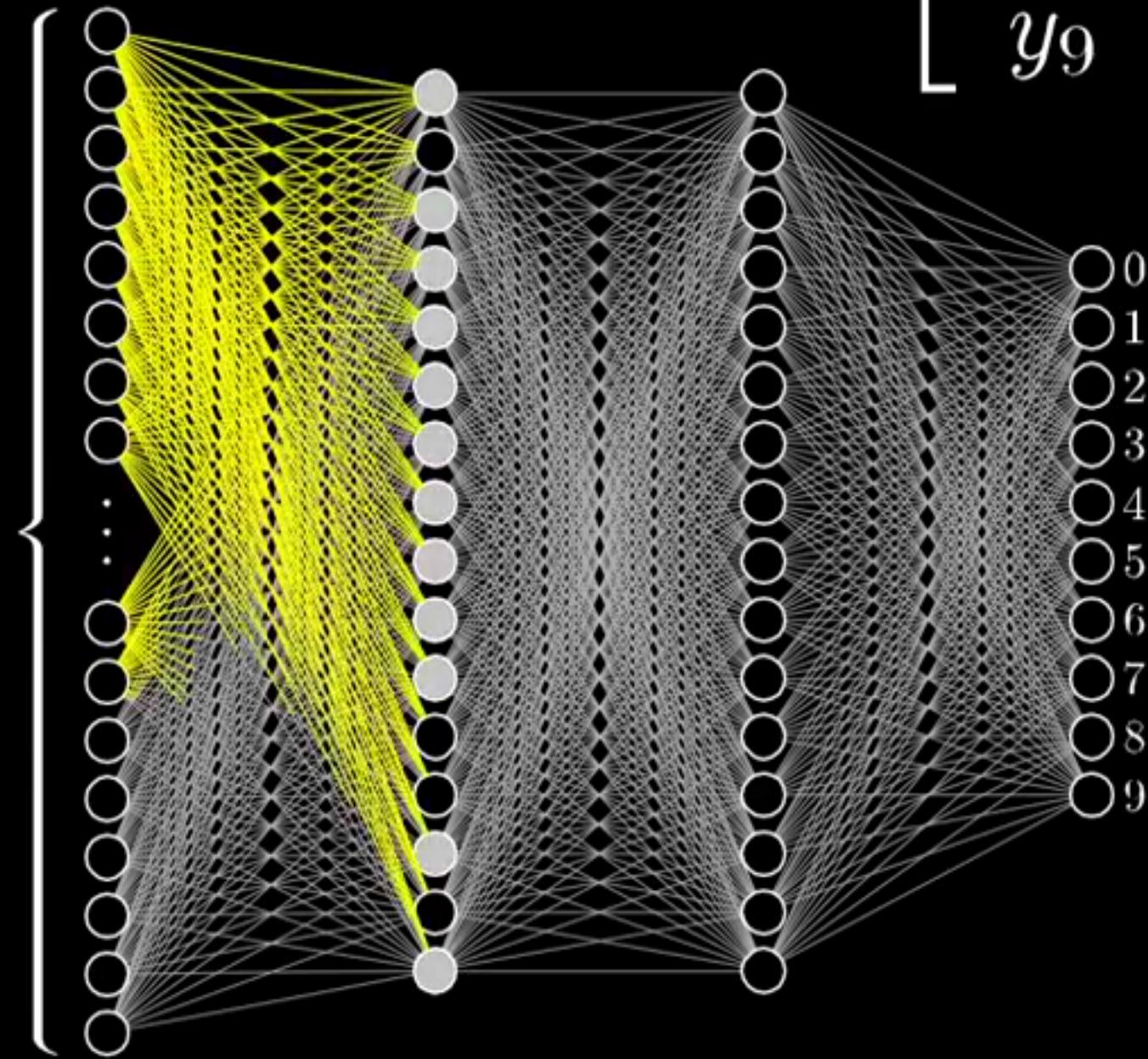


Function

~~Thing that holds
a number~~

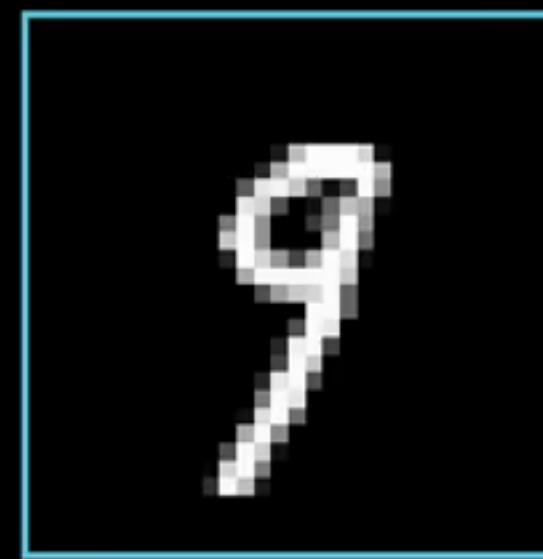


$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$

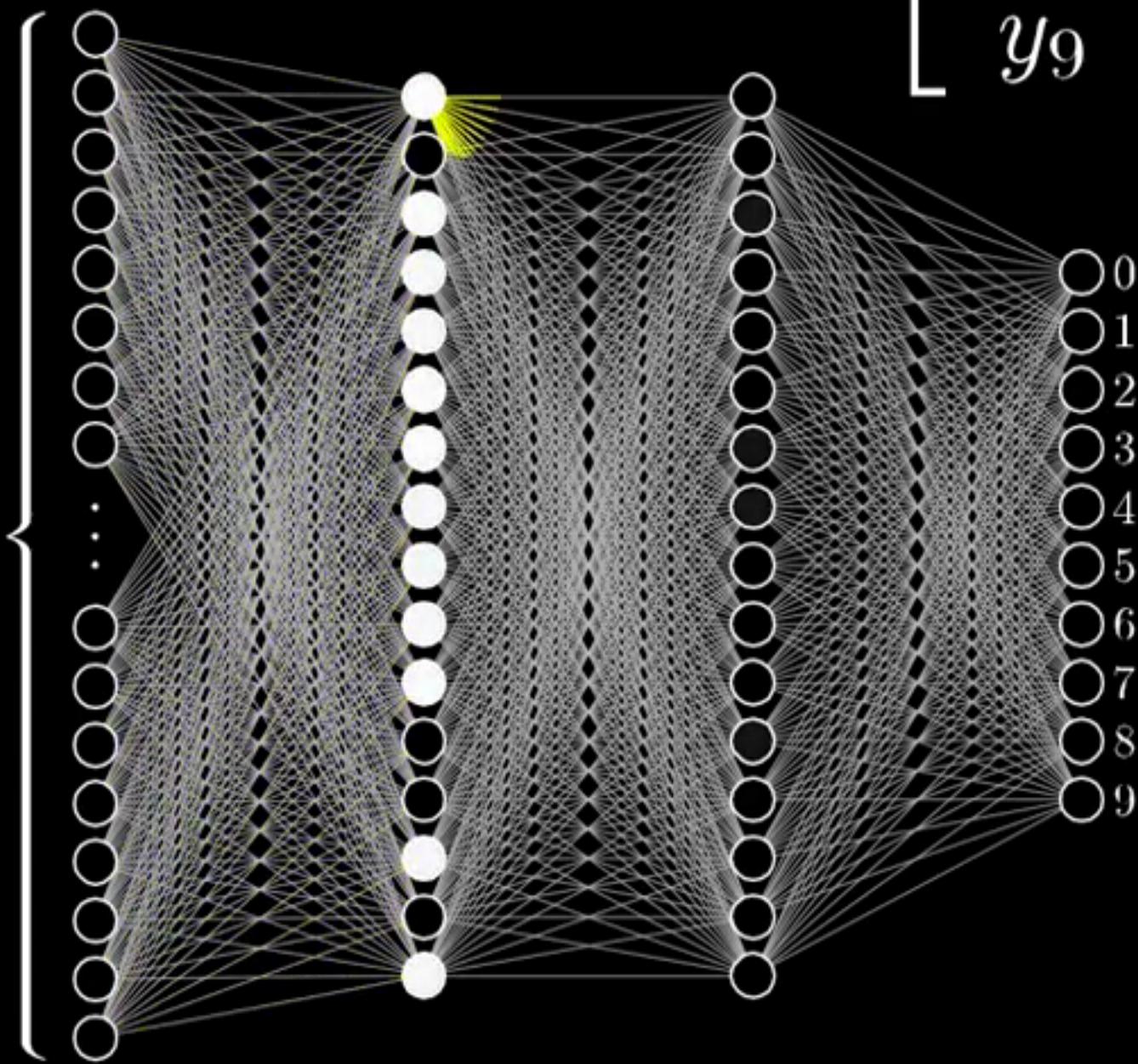


Network
↓
Function

~~Thing that holds
a number~~



784



$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$

Network

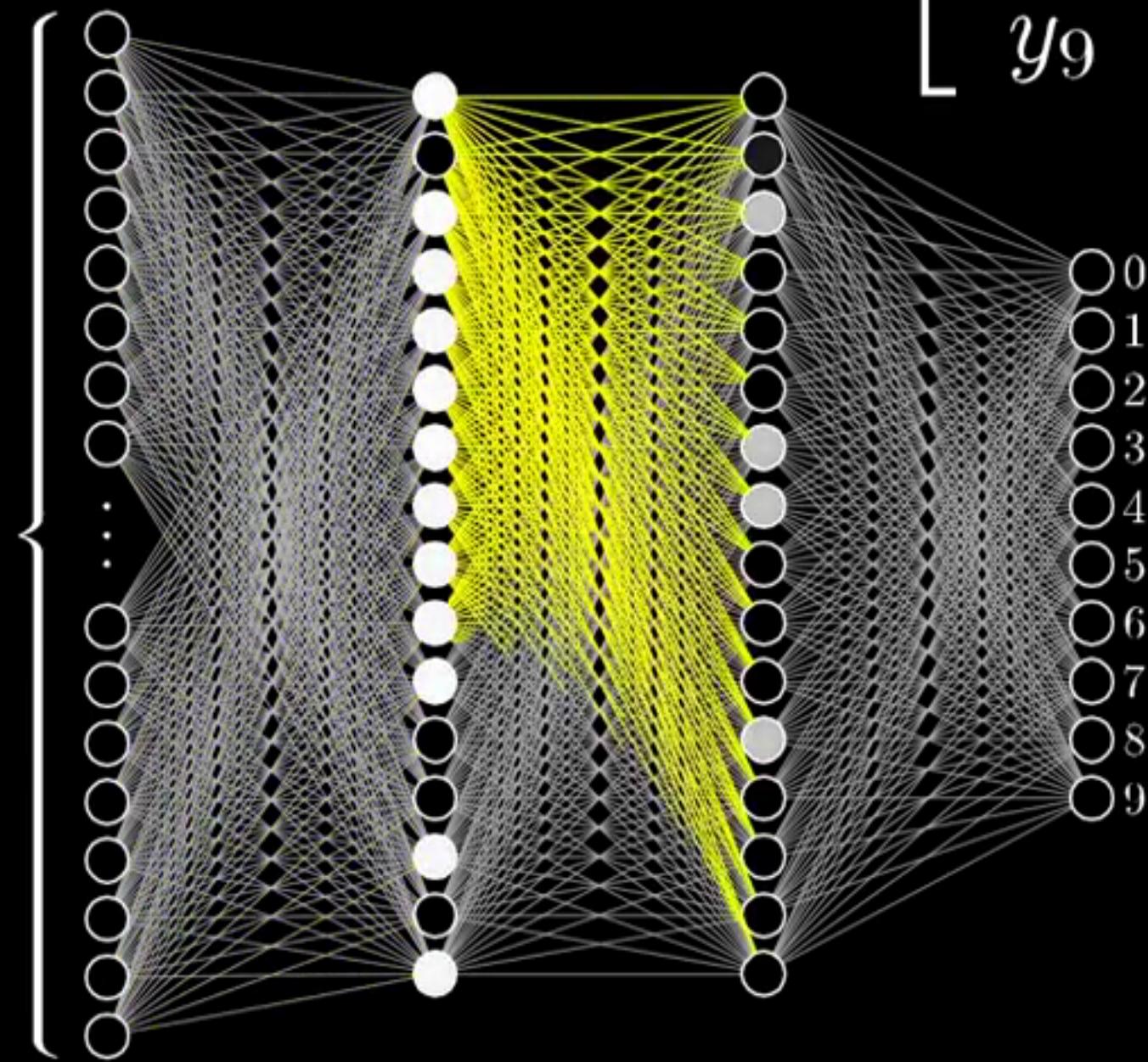


Function

~~Thing that holds
a number~~



$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$

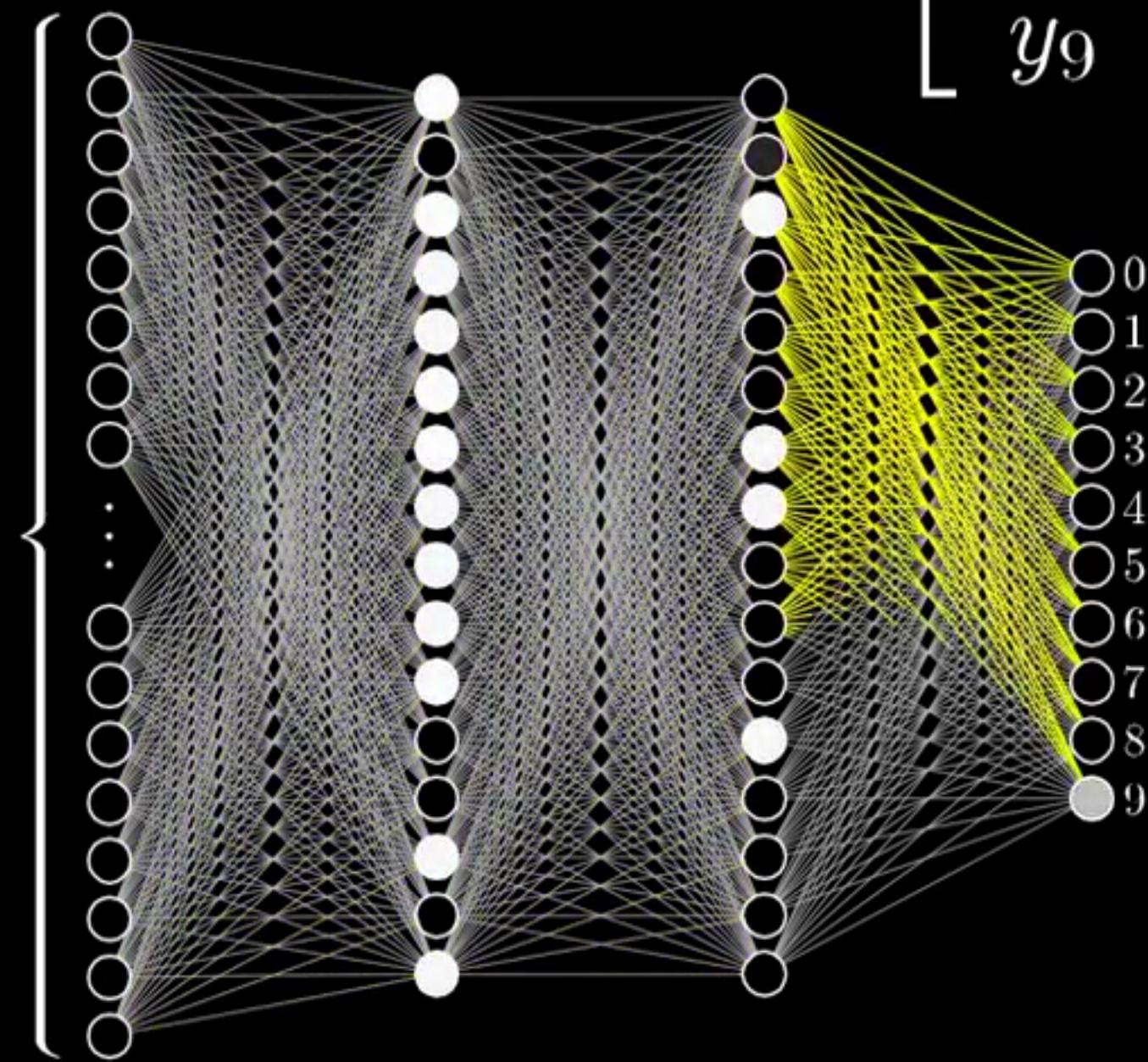


Network
↓
Function

~~Thing that holds
a number~~

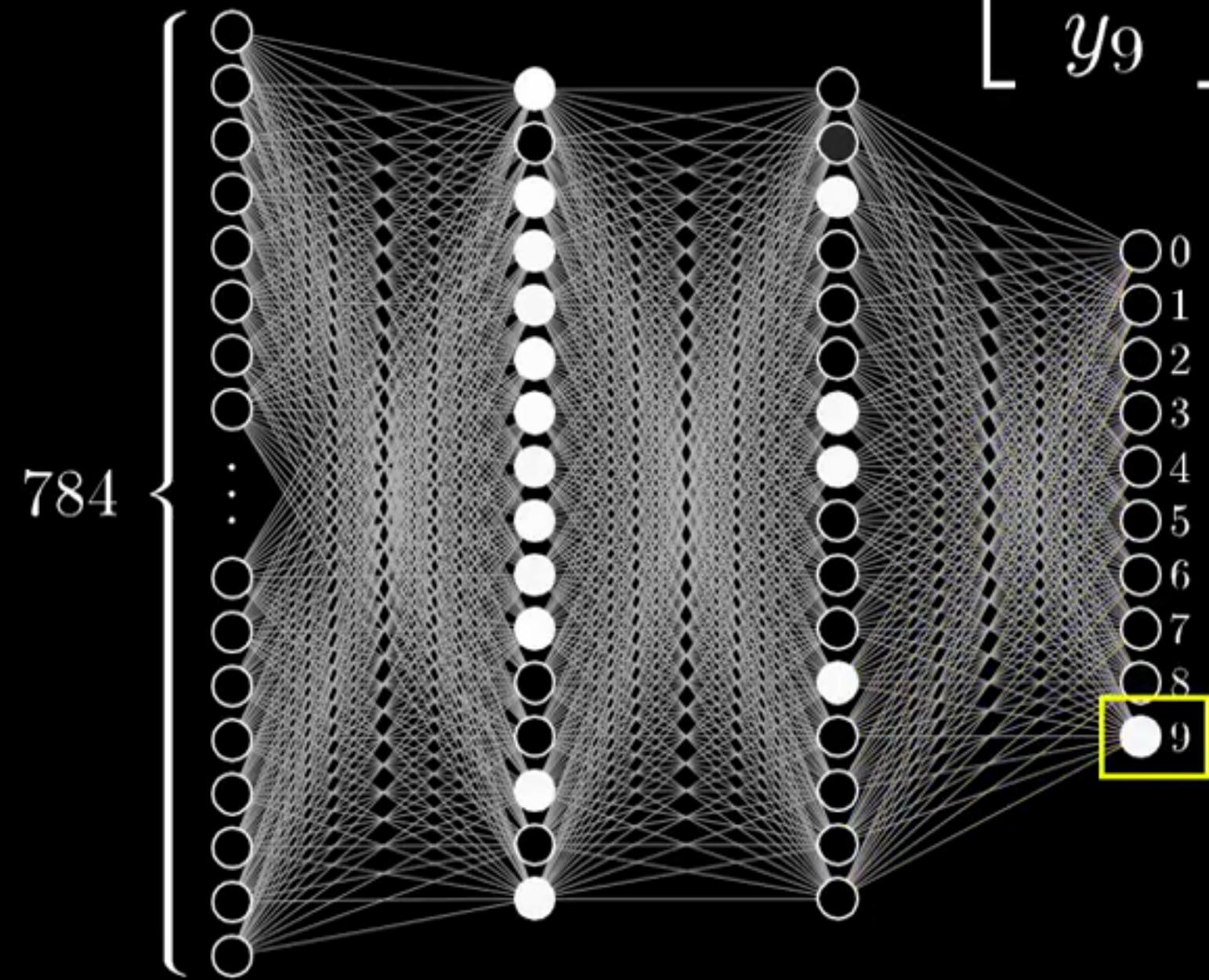


$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$



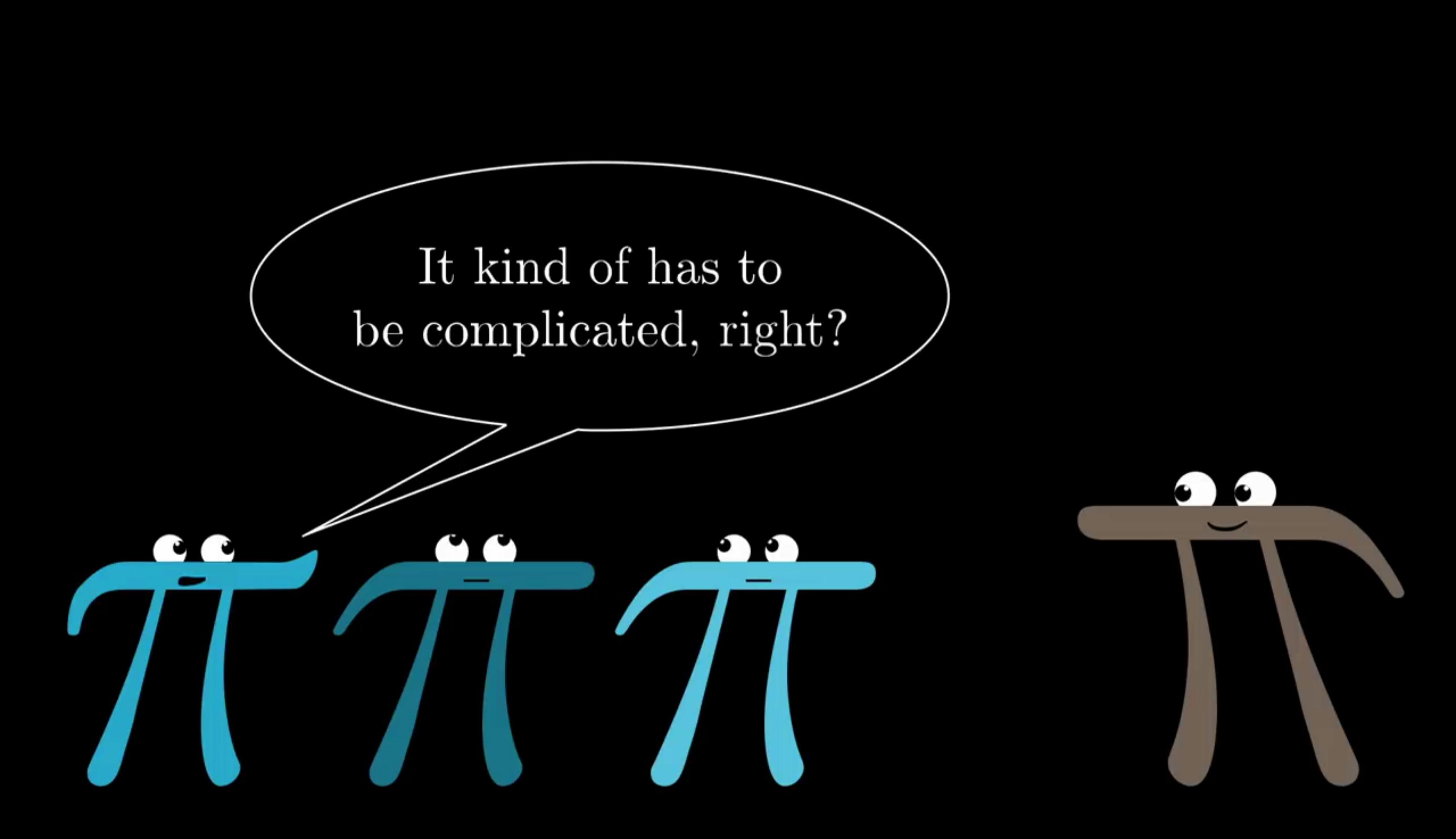
~~Thing that holds
a number~~

$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$

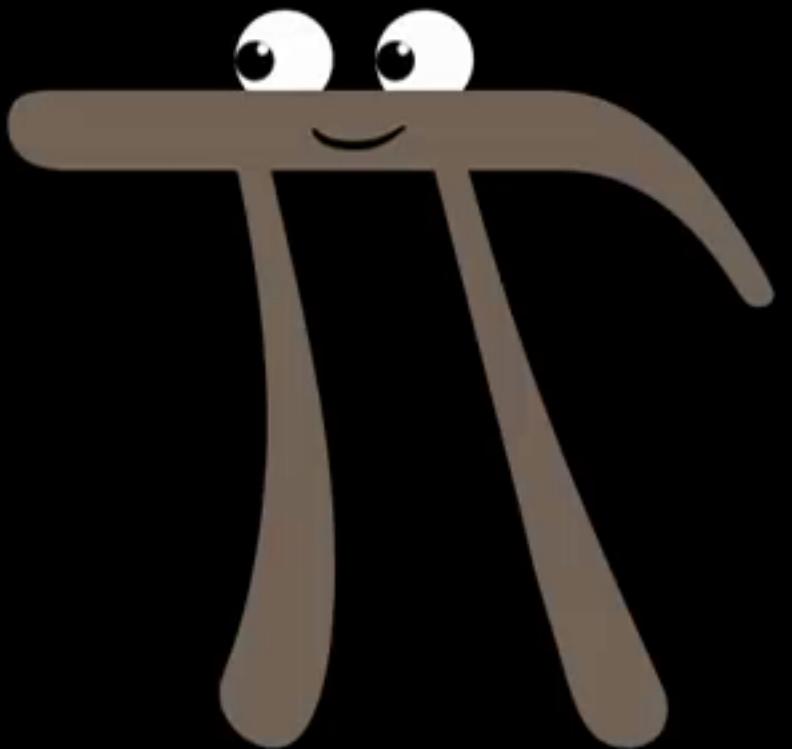
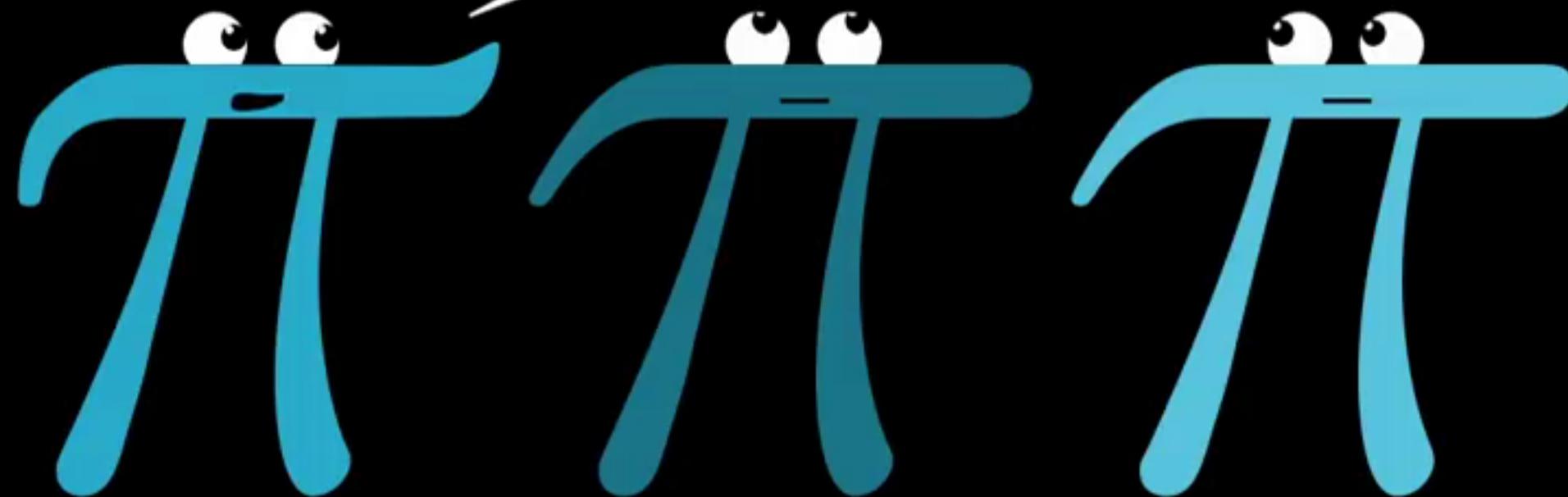


Network
↓
Function

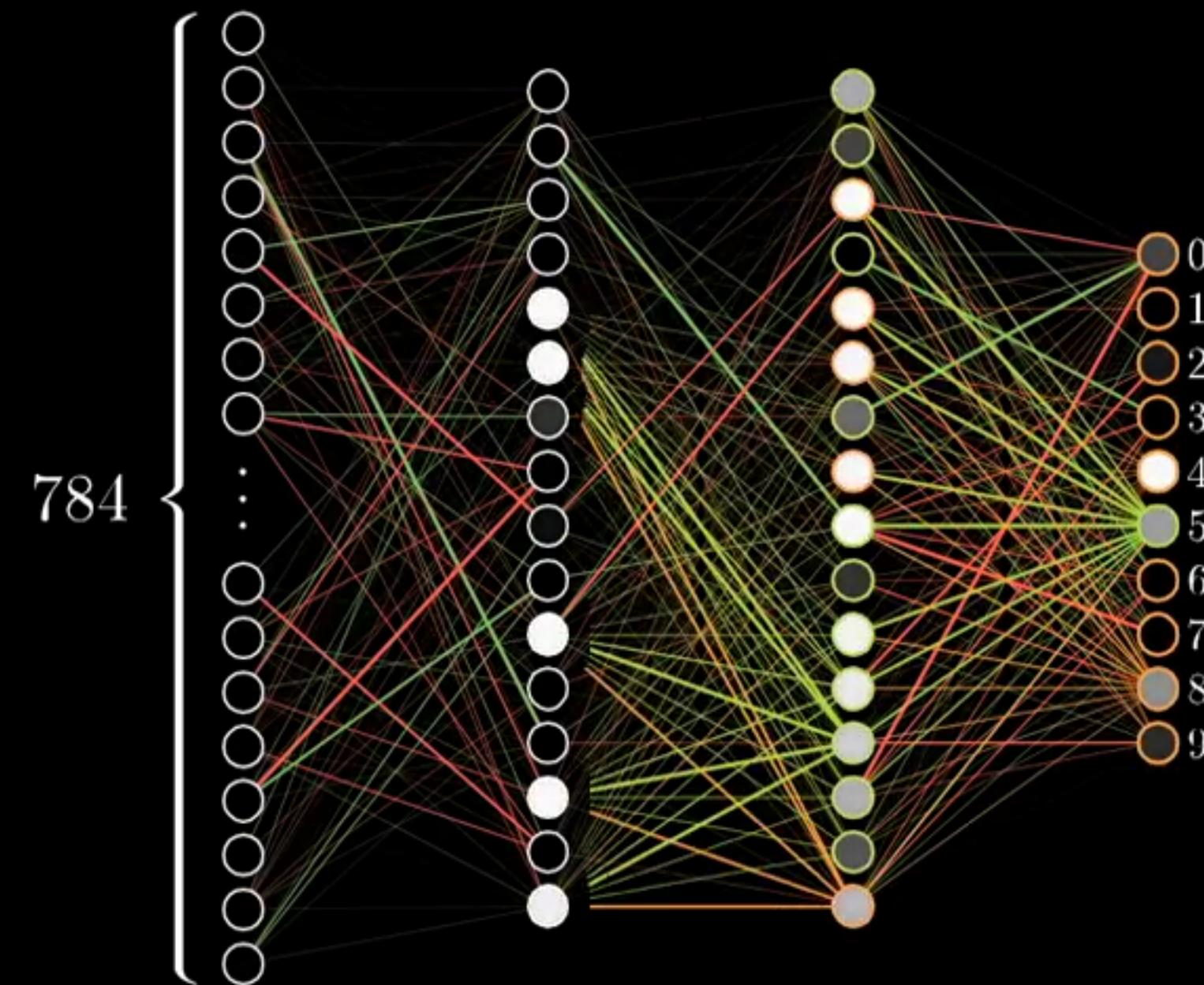
~~Thing that holds
a number~~



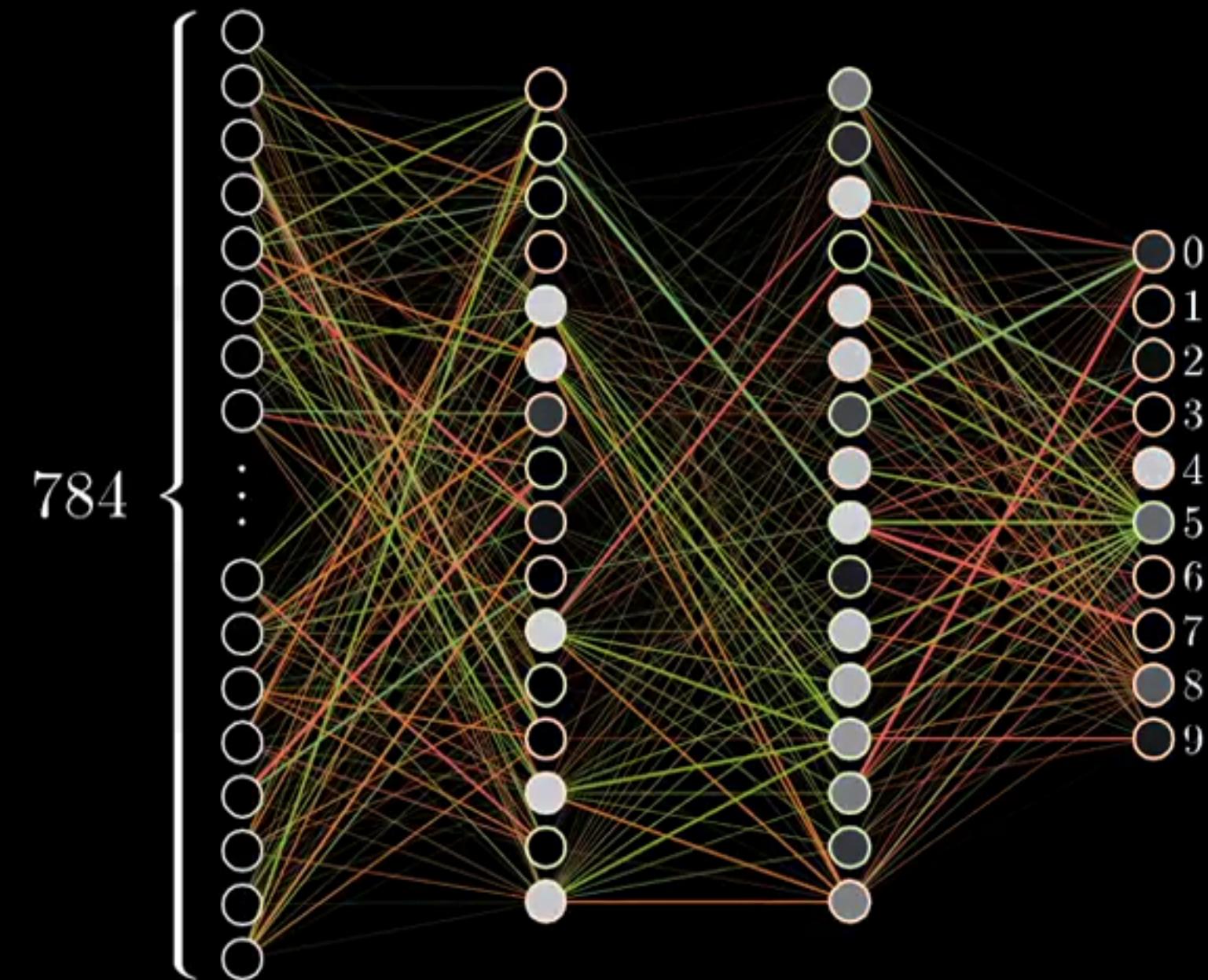
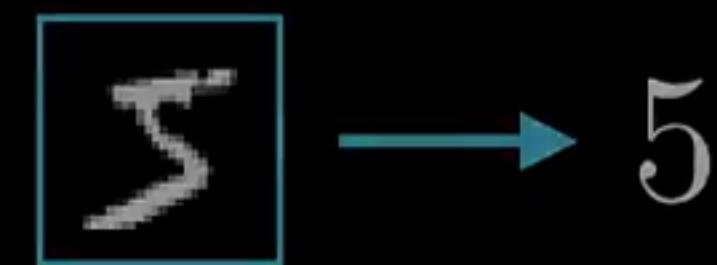
It kind of has to
be complicated, right?



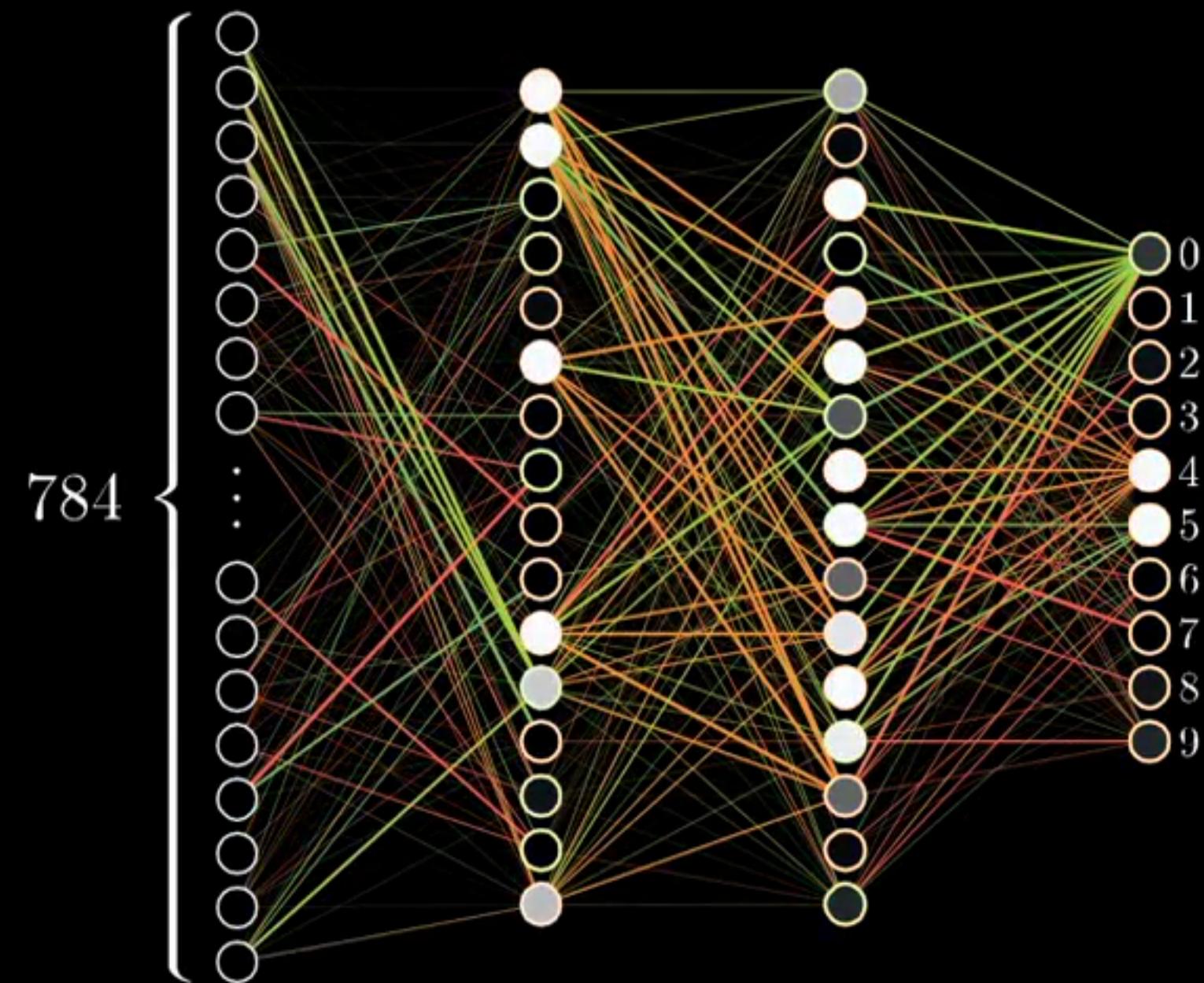
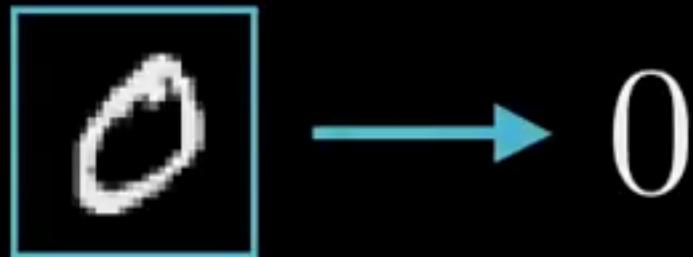
Training in
progress. . .



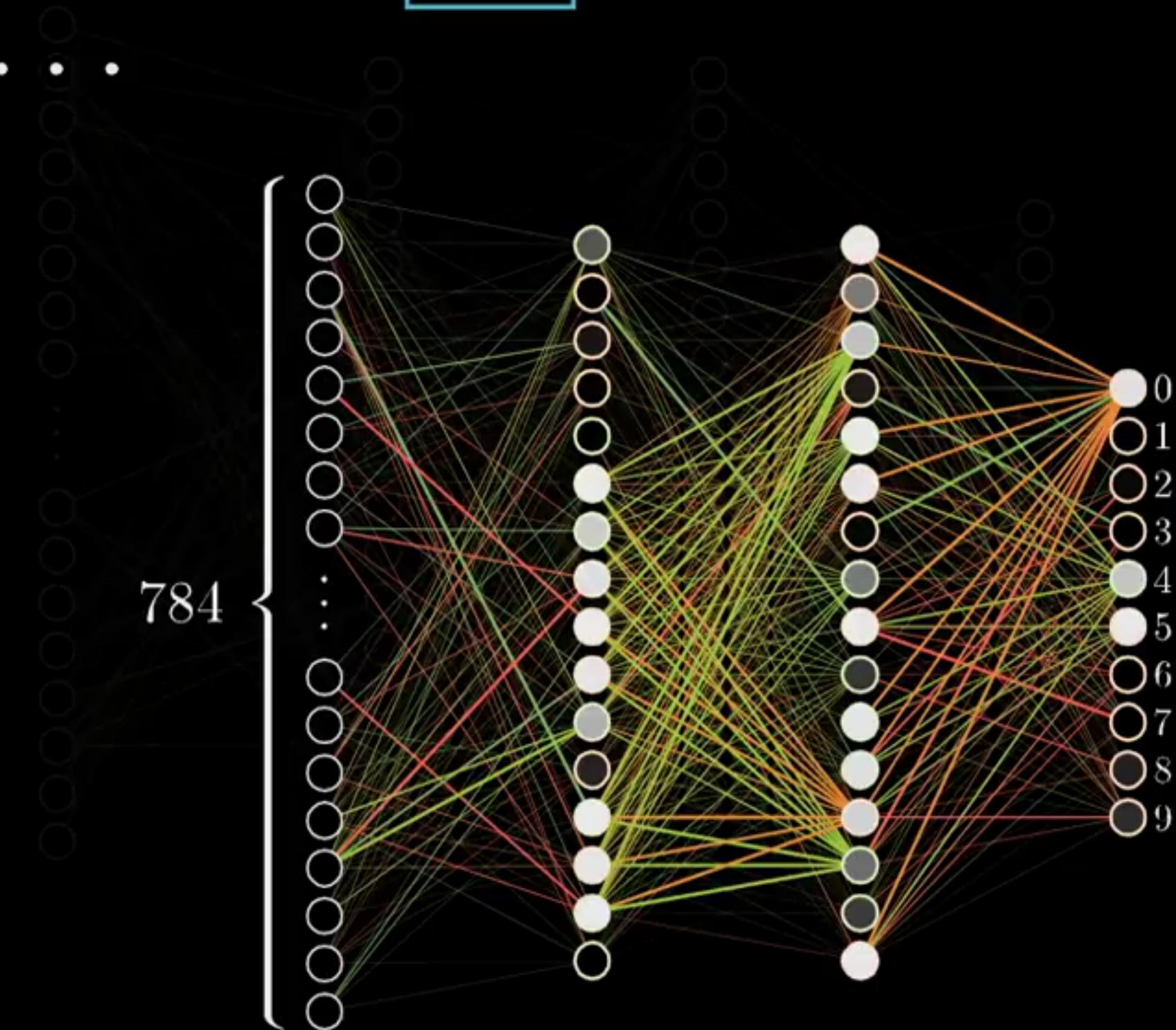
Training in
progress. . .



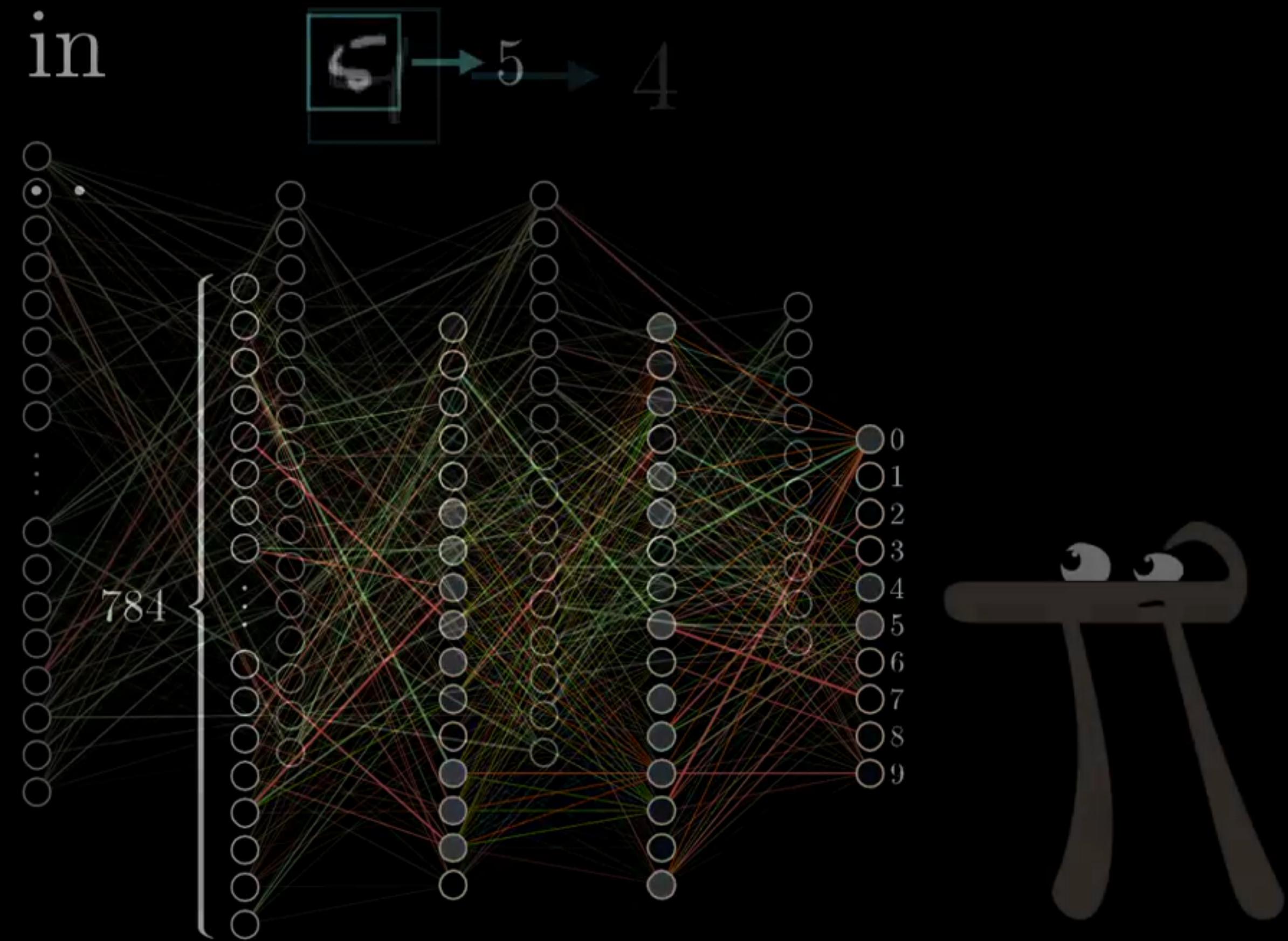
Training in
progress. . .



Training in
progress. . .



Training in progress.

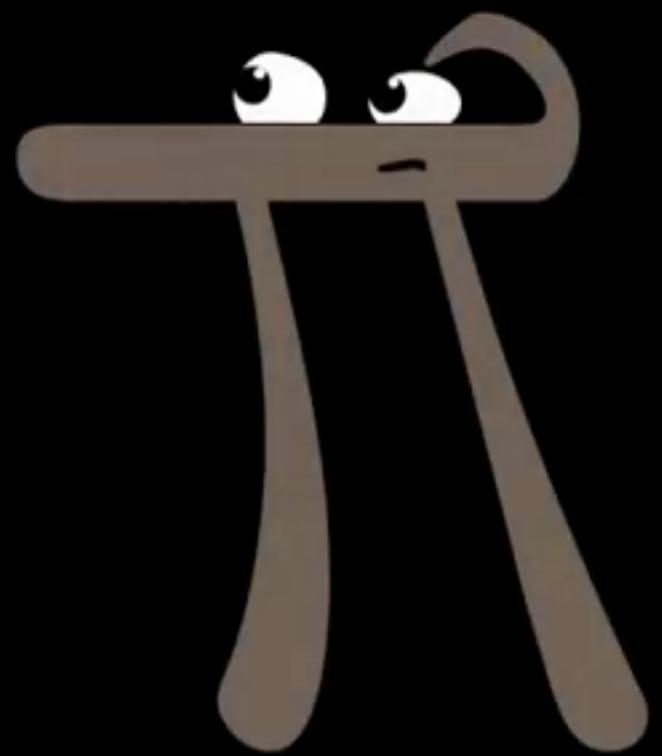
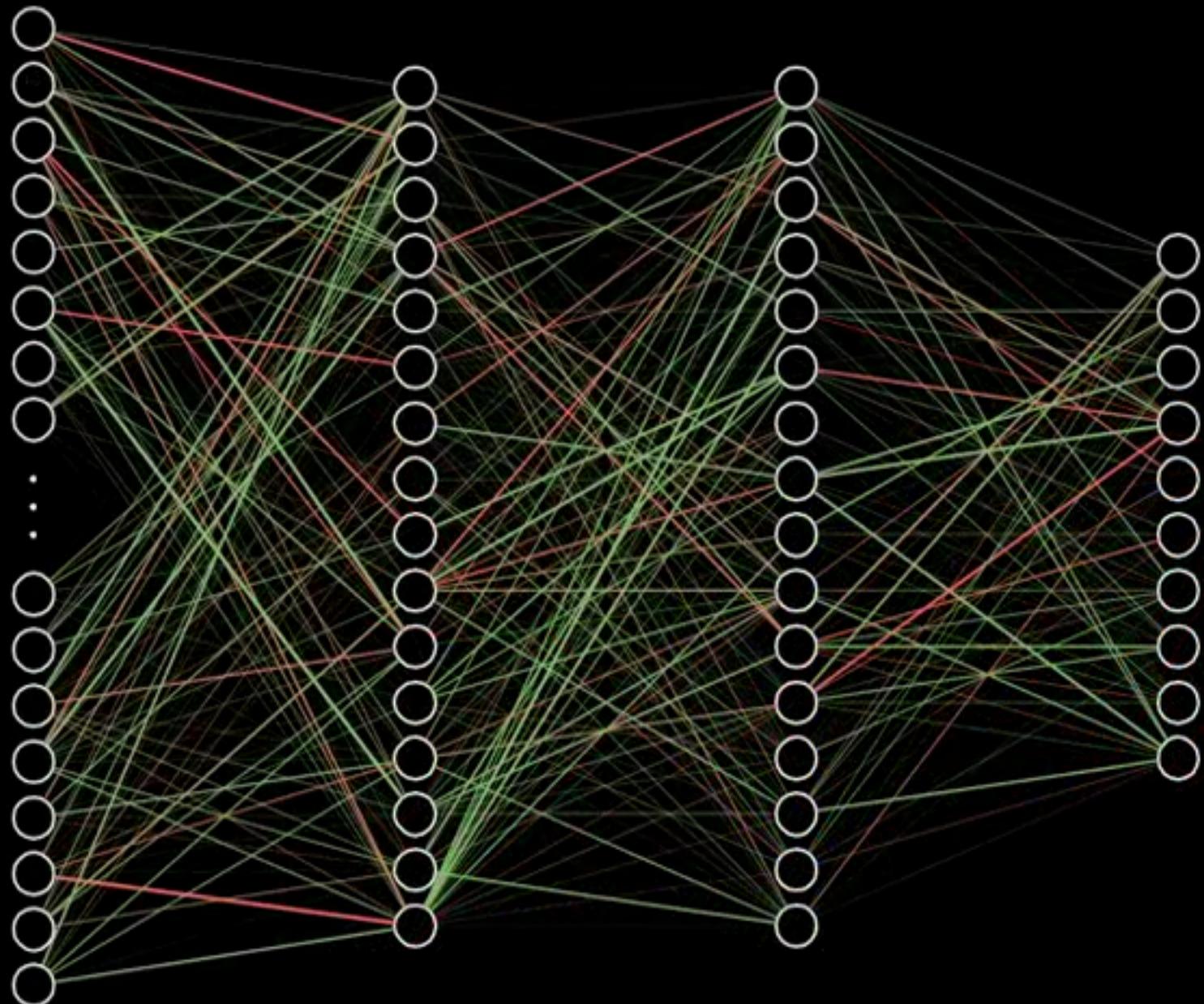


Training in
progress.

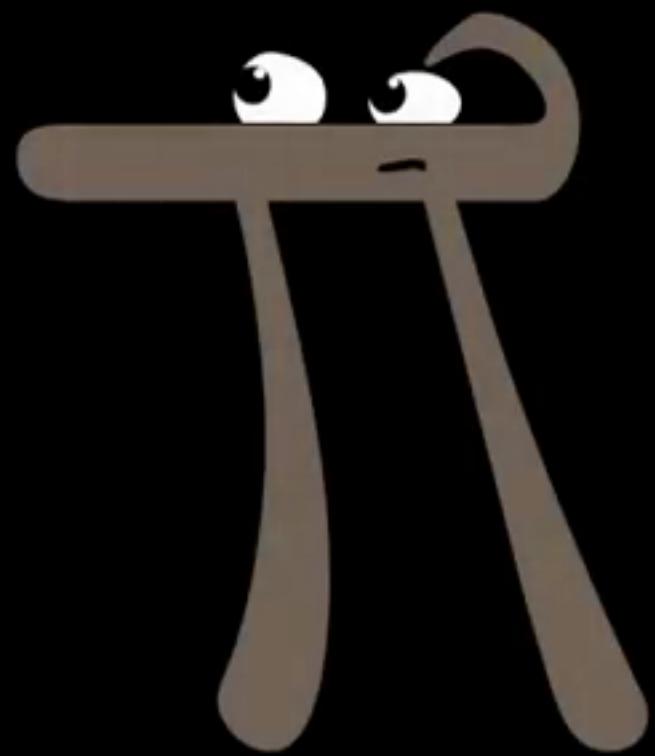
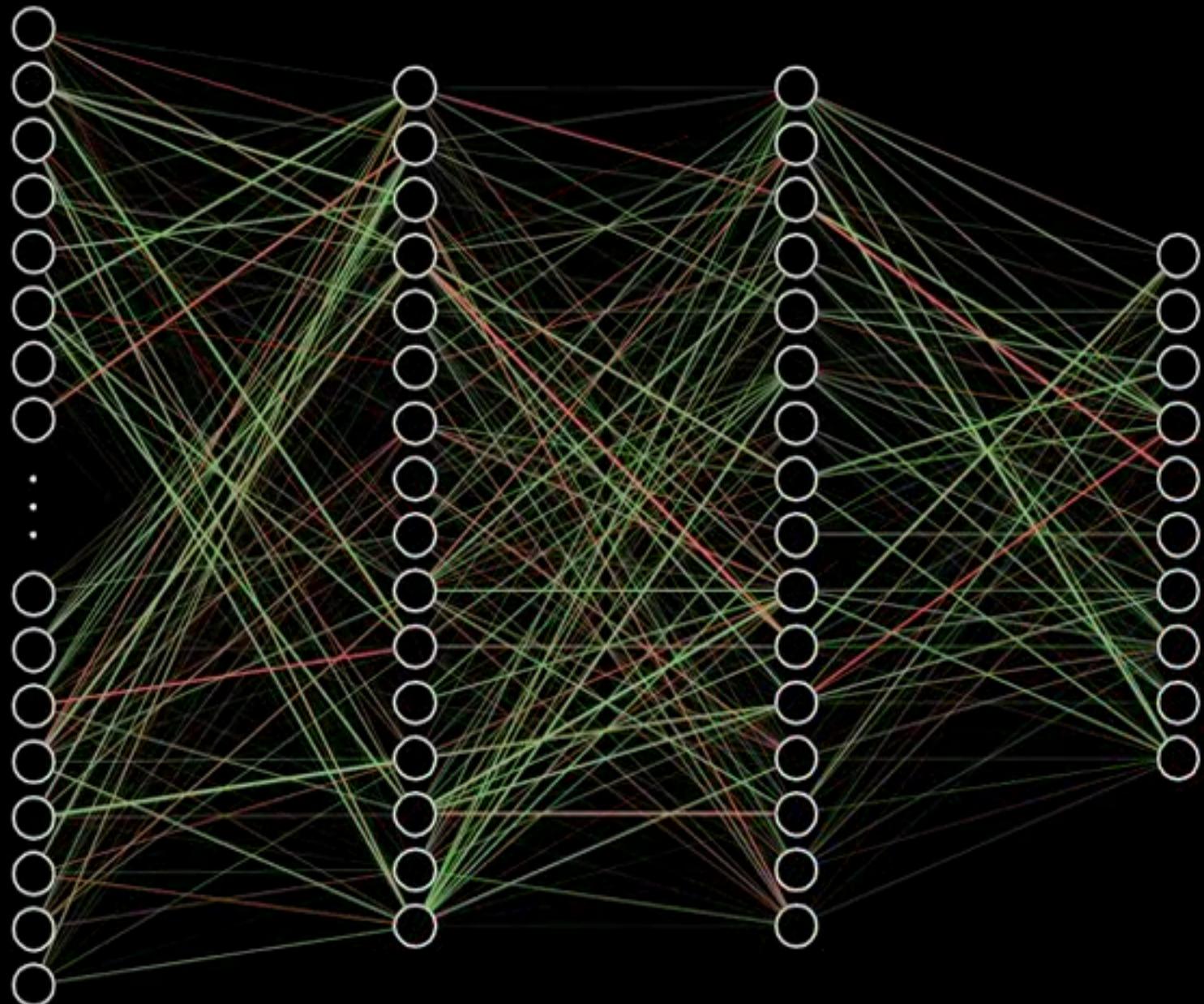
$$q \rightarrow 9$$



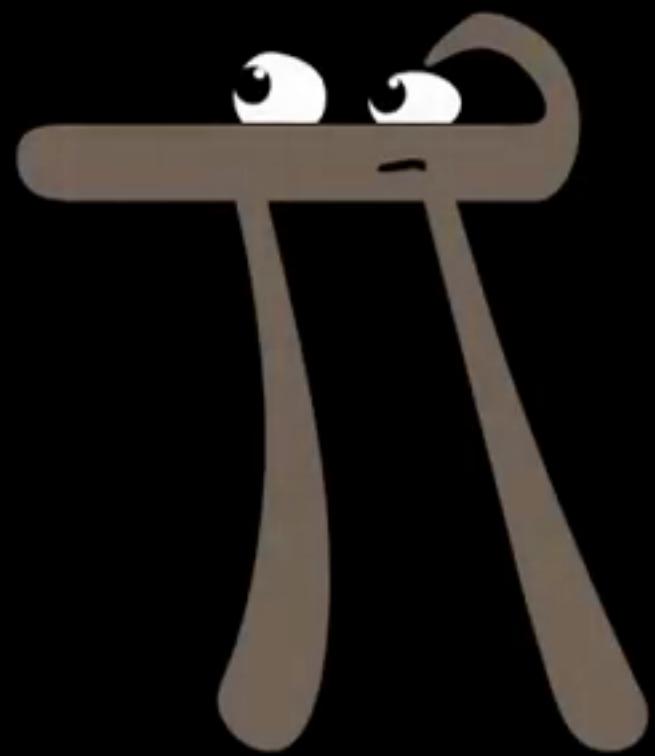
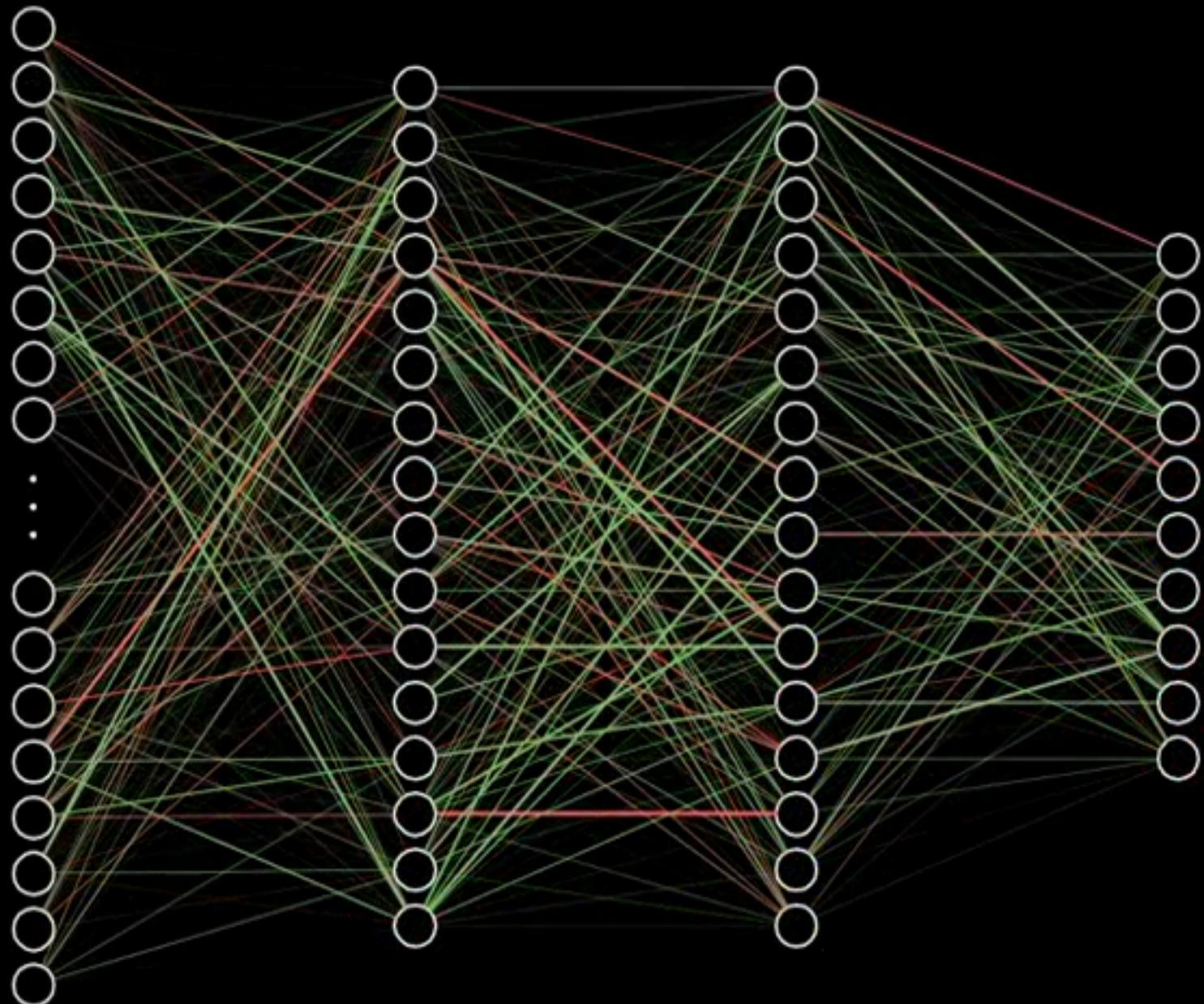
5 → 5



4 → 4



5 → 5



7 → 7

