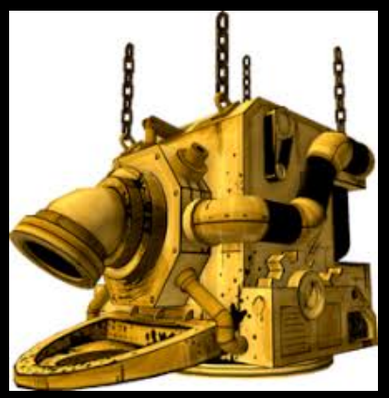
GDG for ML Newcastle

25th Oct. OPEN LAB

What is expected?

- Basic theory of machine learning
- Simple code implementation
- Small tricks in training a model



Machine learning is easy and funny.

Machine learning is the science of getting computers to act without being explicitly programmed.

— Stanford

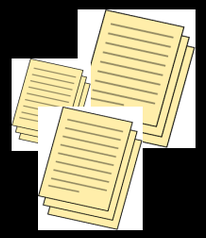
Machine learning is the science of getting computers to act without being explicitly programmed. But instead letting them learn a few tricks on their own.

— Stanford, Max-Planck Institute.

Machine learning is the science of getting computers to act without being explicitly programmed. But instead letting them learn a few tricks on their own.

— Stanford, Max-Planck Institute.

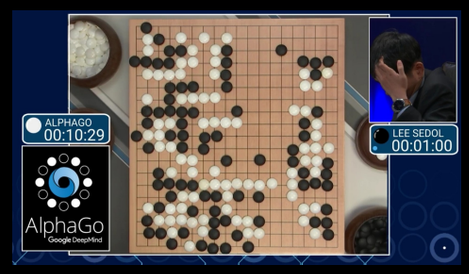
Observations



Model

Outcomes

Dog/Cat



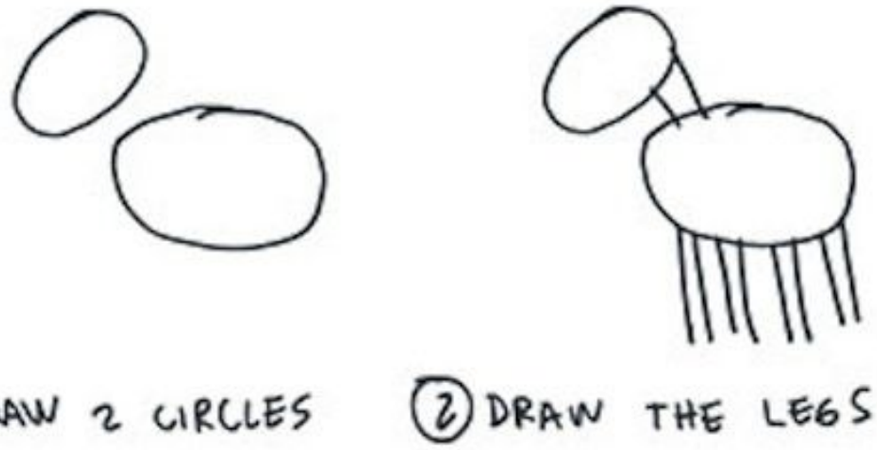
https://developers.google.com/machine-learning/crash-course/

Build a Model

- Representation
  - LBP, SIFT.
- Modelling
  - Classification: Cross entropy
- Optimisation
  - Gradient Decent, Expectation Maximisation

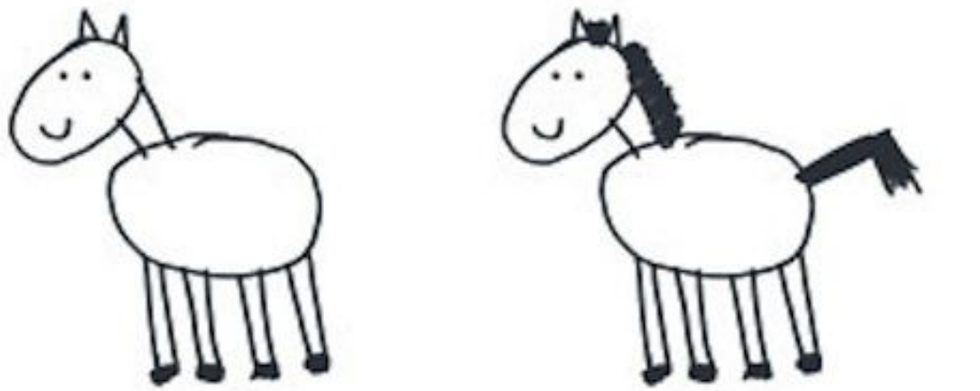
HOW TO:
DRAW A HORSE

BY VAN OKTOP



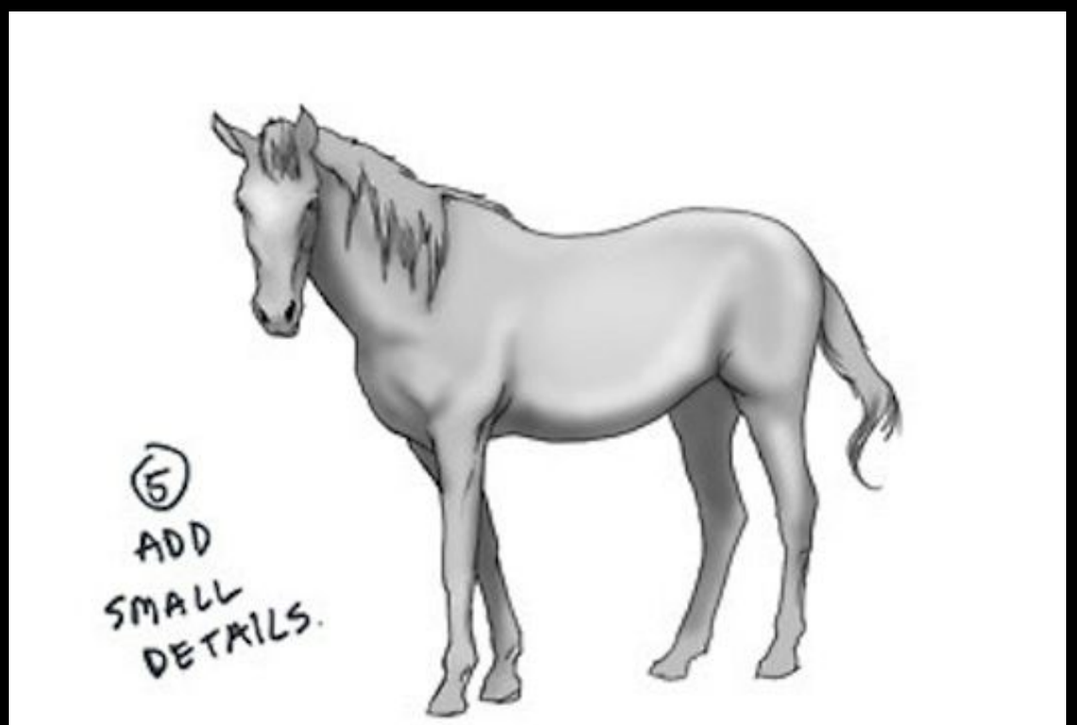
① DRAW 2 CIRCLES

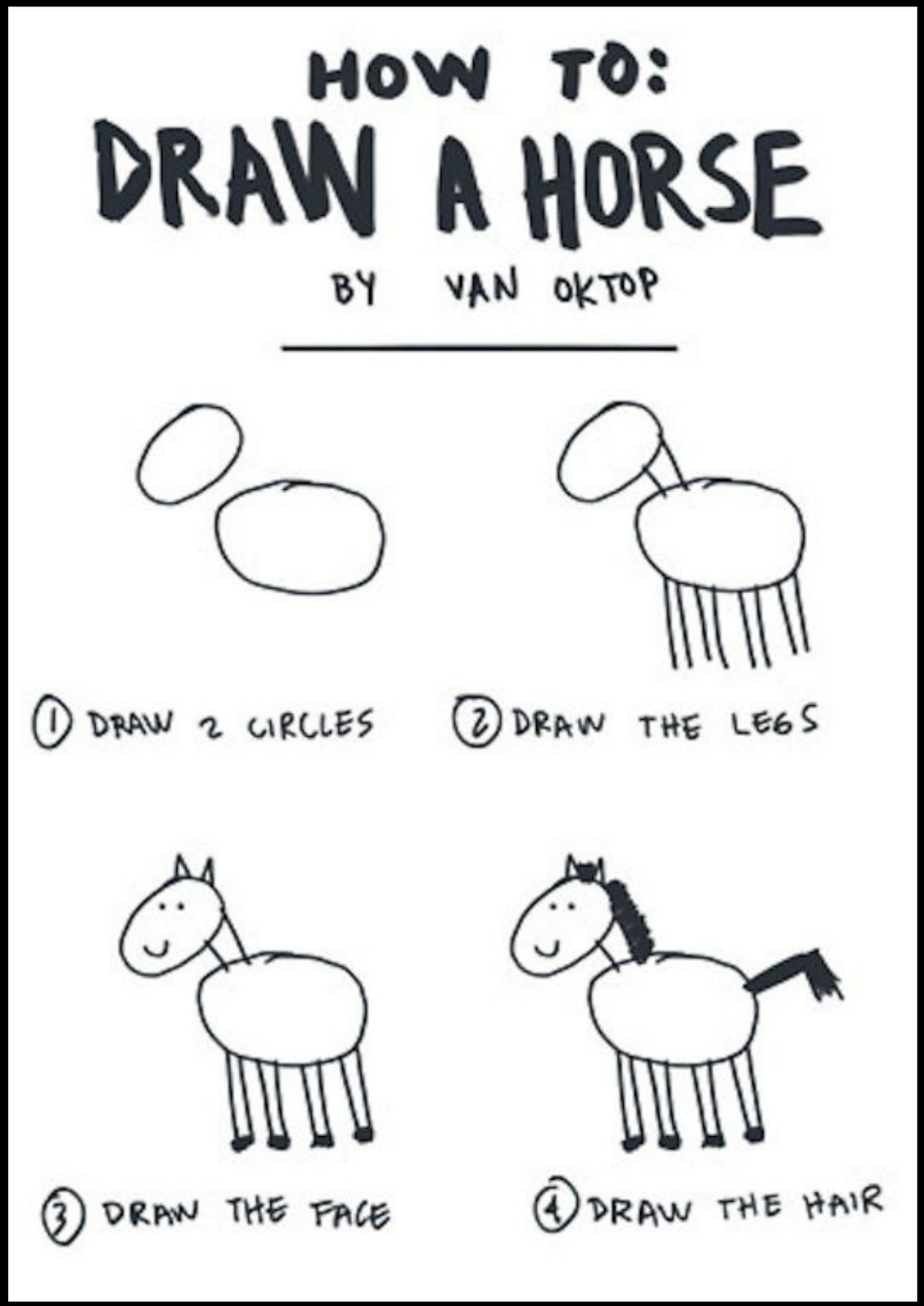
② DRAW THE LEGS

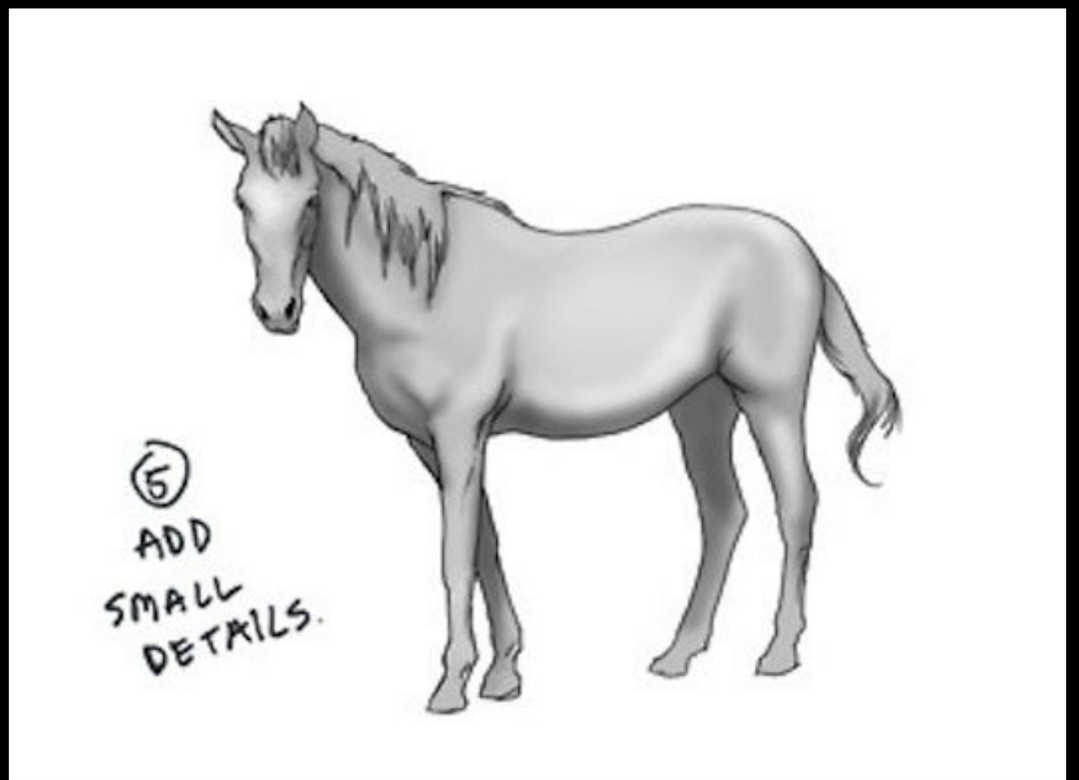


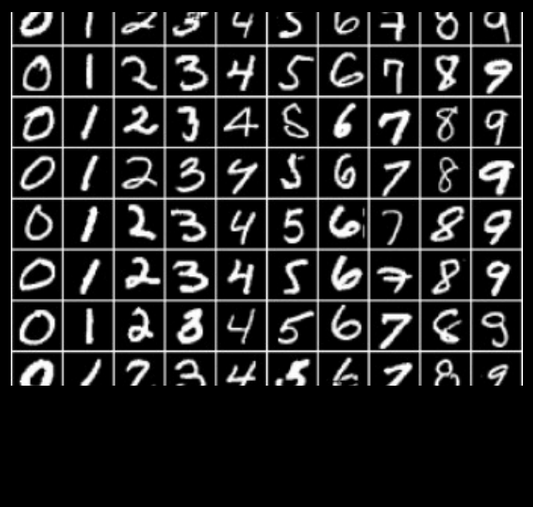
③ DRAW THE FACE

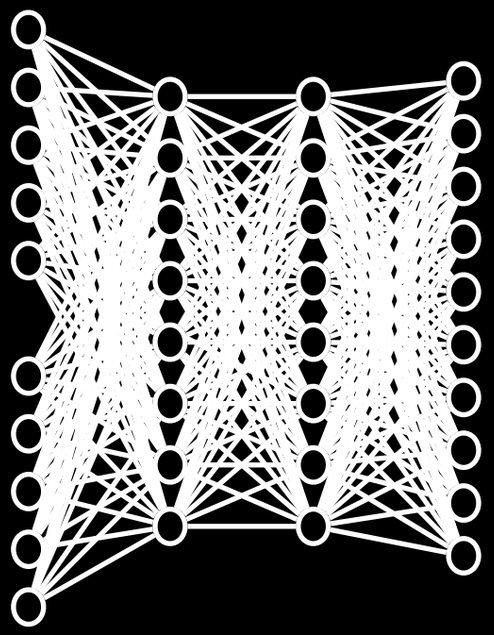
④ DRAW THE HAIR




HOW TO:
DRAW A HORSE
BY VAN OKTOP
① DRAW 2 CIRCLES
② DRAW THE LEGS
③ DRAW THE FACE
④ DRAW THE HAIR


⑤
ADD
SMALL
DETAILS.





0 ?

1

2

3

4

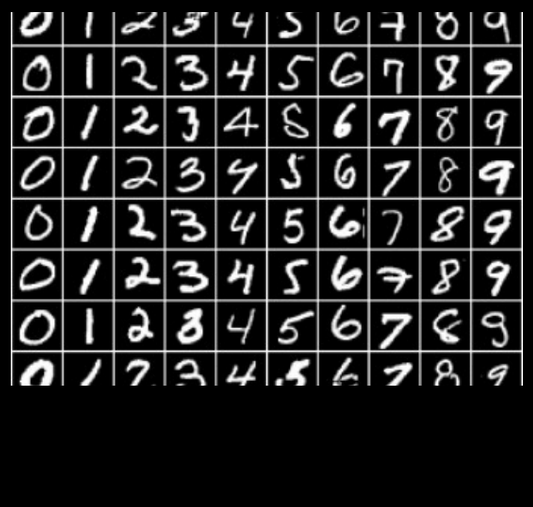
5

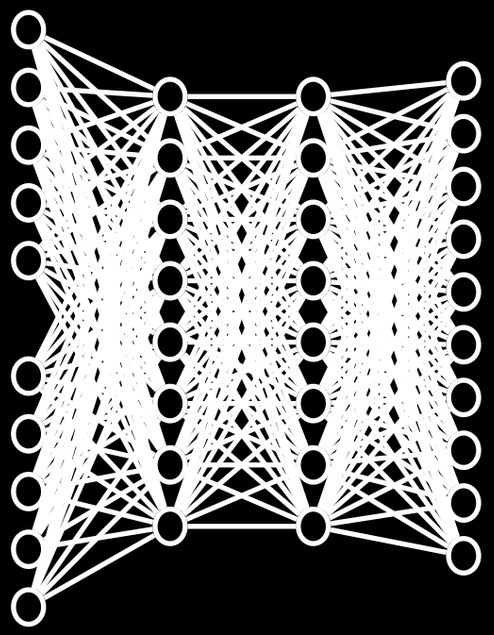
6

7

8

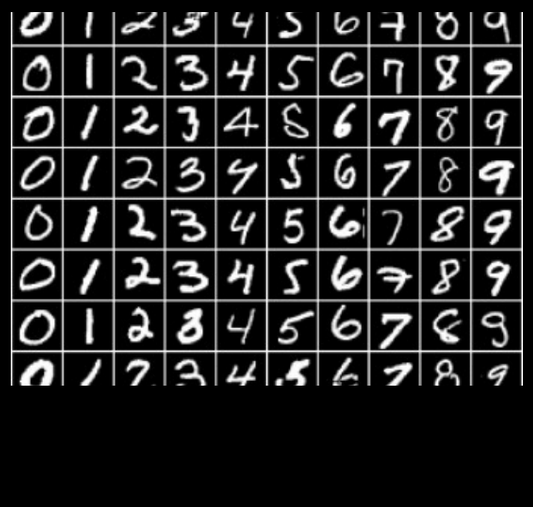
9

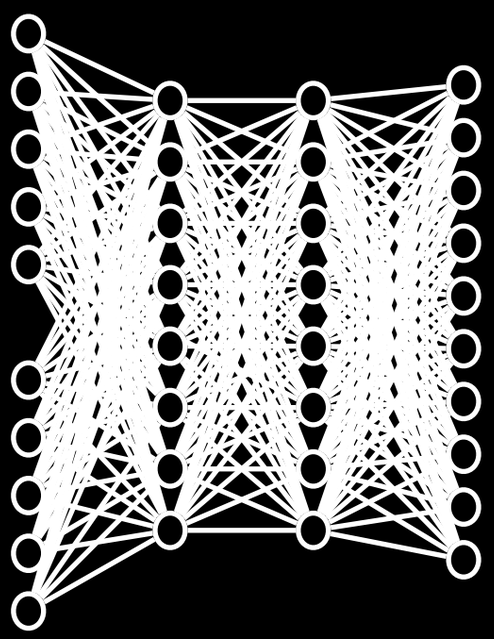




0
1
2
3
4
5
6
7
8
9

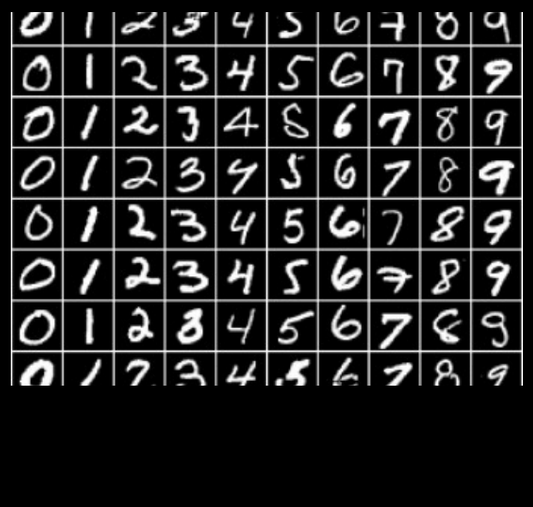
?

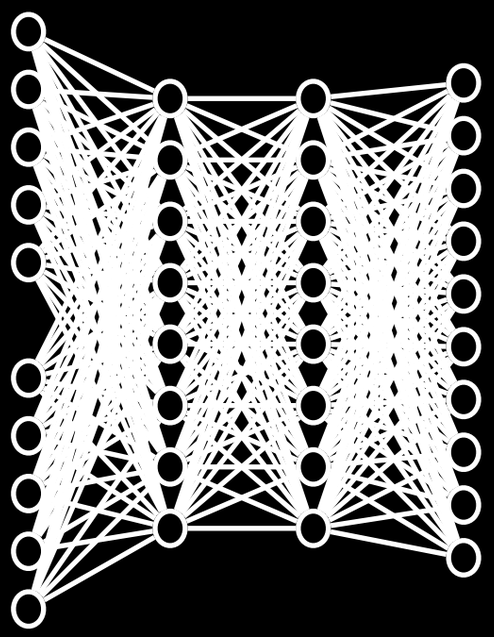




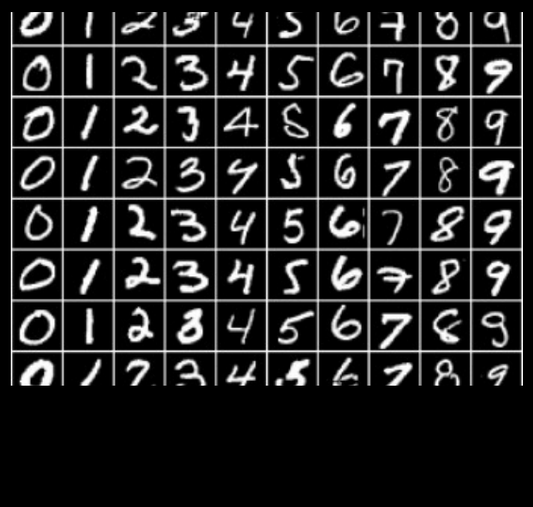
0
1
2
3
4
5
6
7
8
9

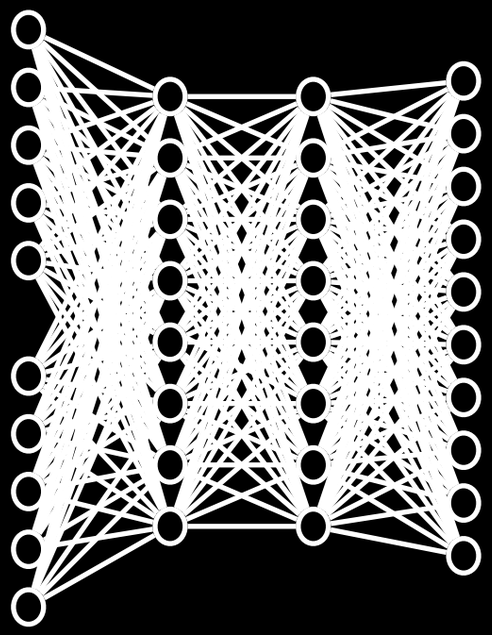
?





0
1
2
3
?
4
5
6
7
8
9





0

1

2

3

4

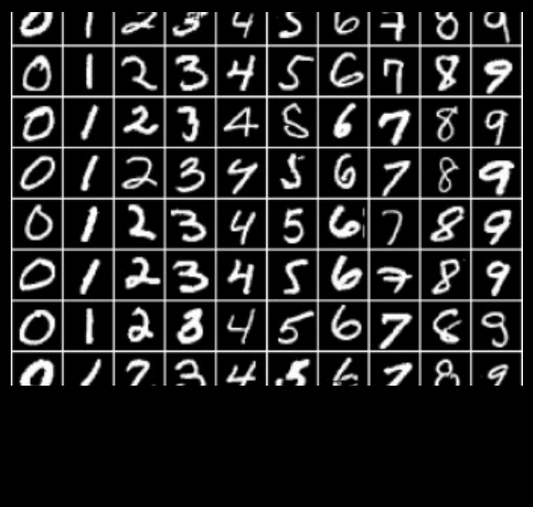
5

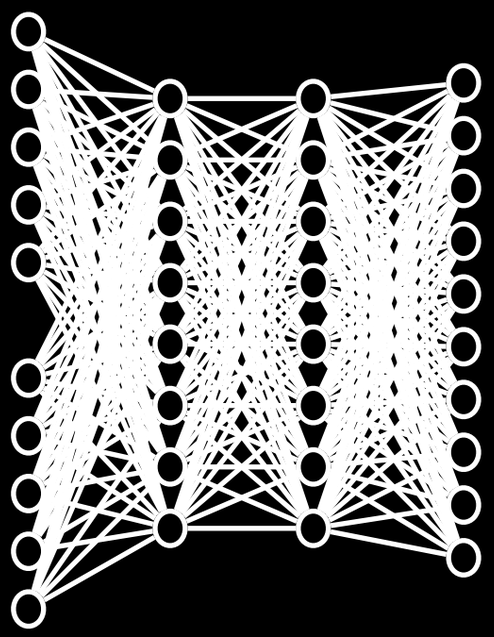
6

7

8 ?

9





0

1

2

3

4

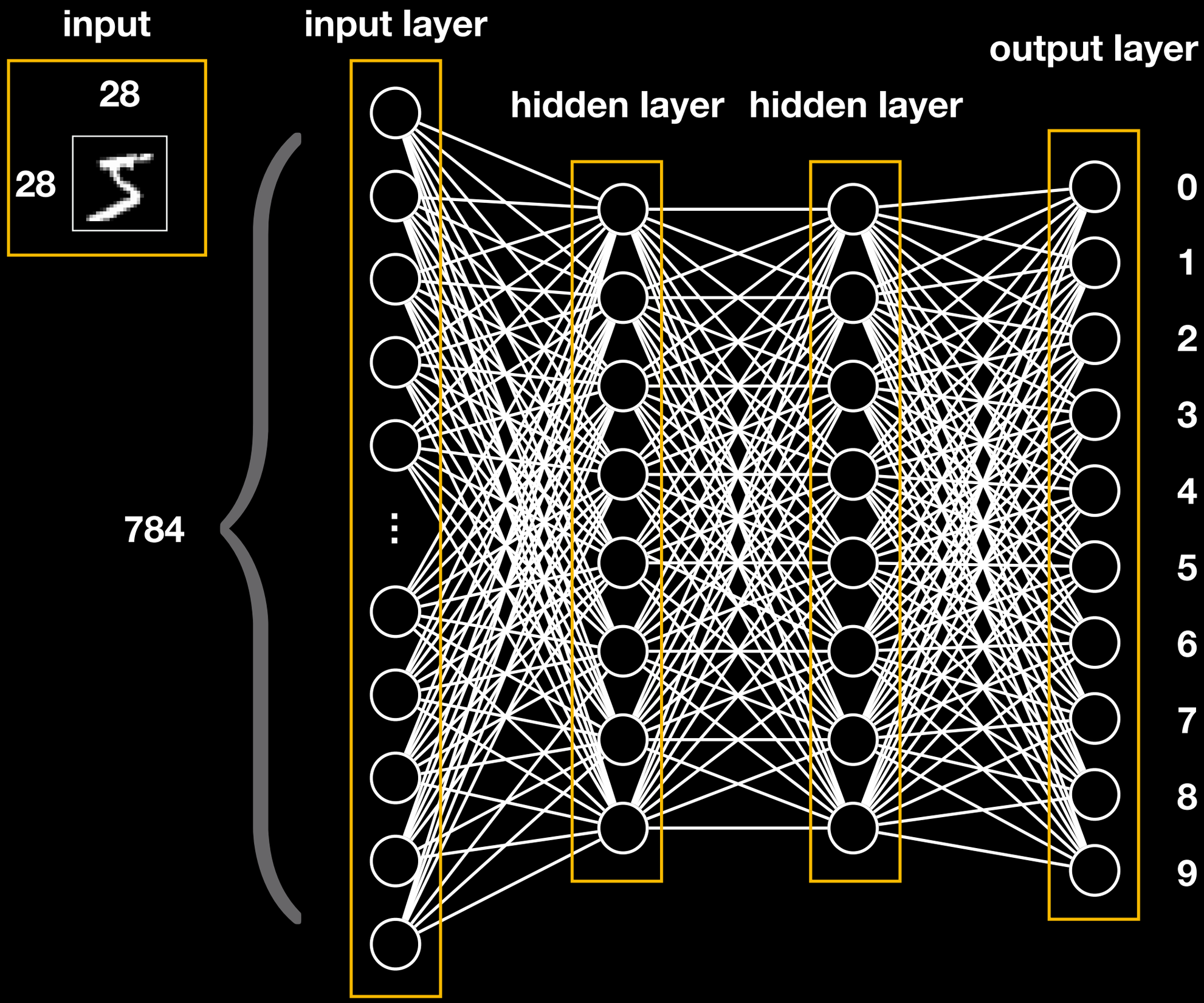
5

6

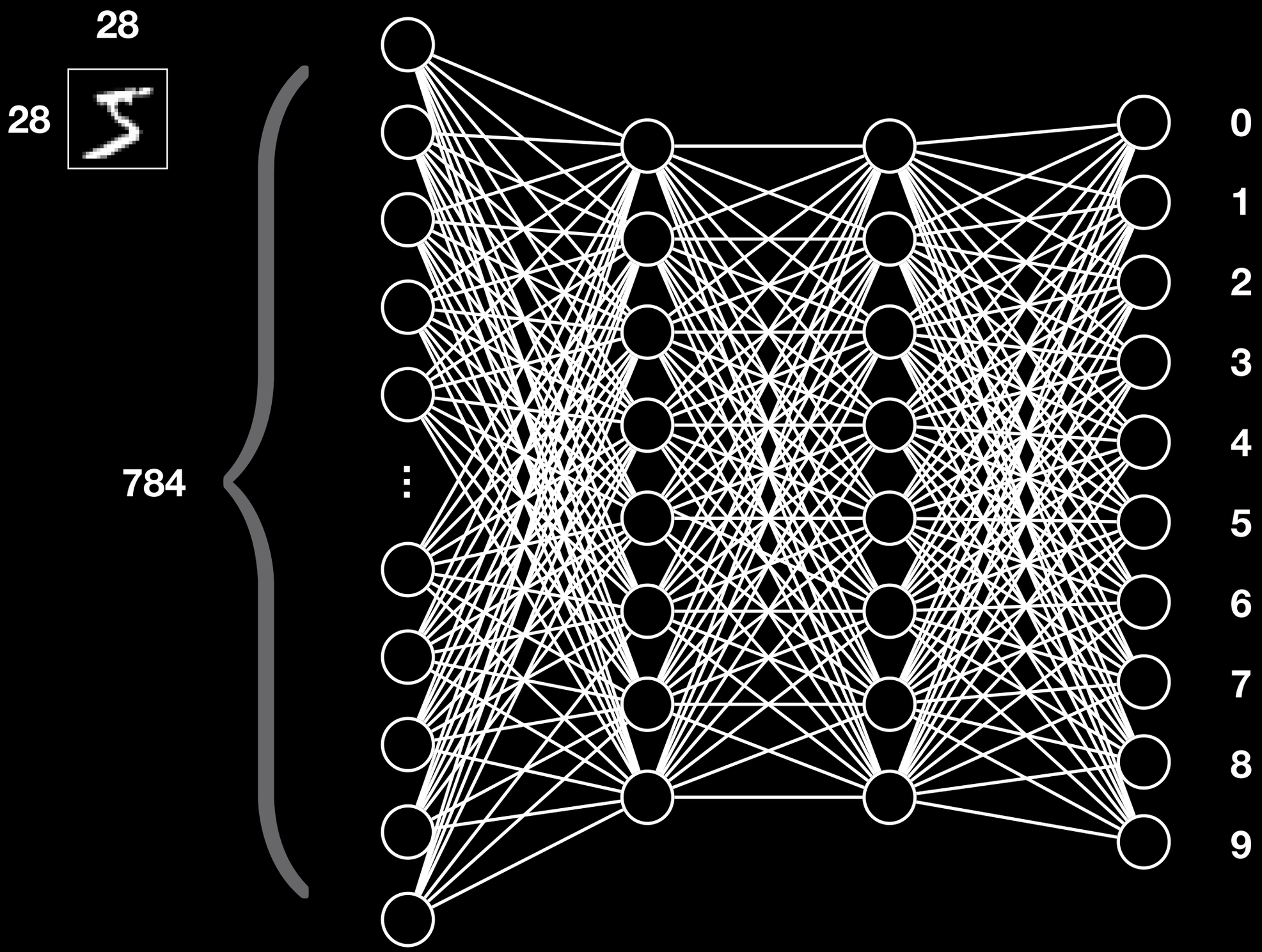
7

8

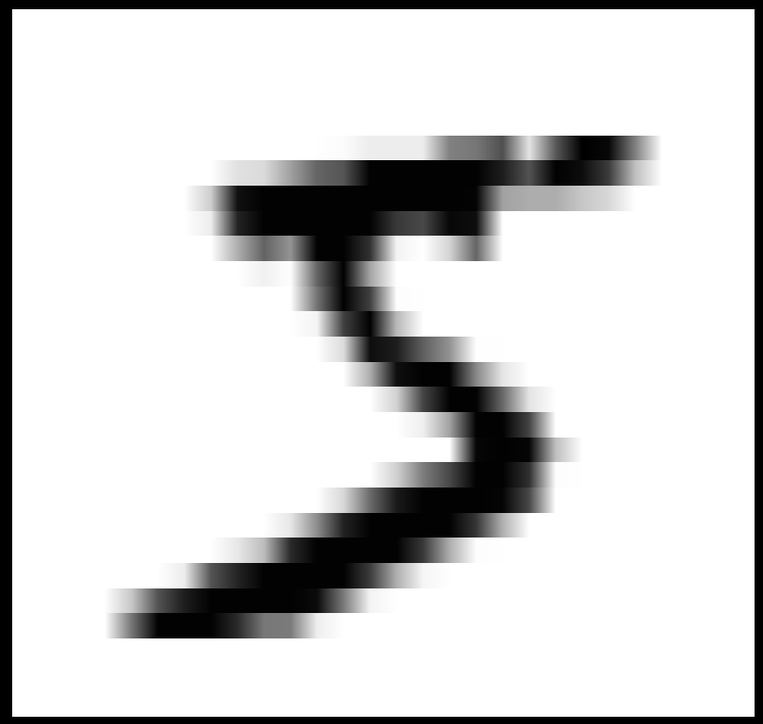
9 ?


input
input layer
hidden layer
hidden layer
output layer
28
28
784
0
1
2
3
4
5
6
7
8
9
Neurons: Placeholders that take the input.
Connections: Parameters/Weights of the network.

					18	18	18	126	136	175	26	166	255	247	127
	30	36	94	154	170	253	253	253	253	253	225	172	253	242	195	64
49	238	253	253	253	253	253	253	253	253	253	251	93	82	82	56	39
18	219	253	253	253	253	253	198	182	247	241						
	80	156	107	253	253	205	11	0	43	154						
		14	1	154	253	90										
				139	253	190	2									
				11	190	253	70									
					35	241	225	160	108	1						
						81	240	253	253	119	25					
							45	186	253	253	150	27				
								16	93	252	253	187				
										249	253	249	64			

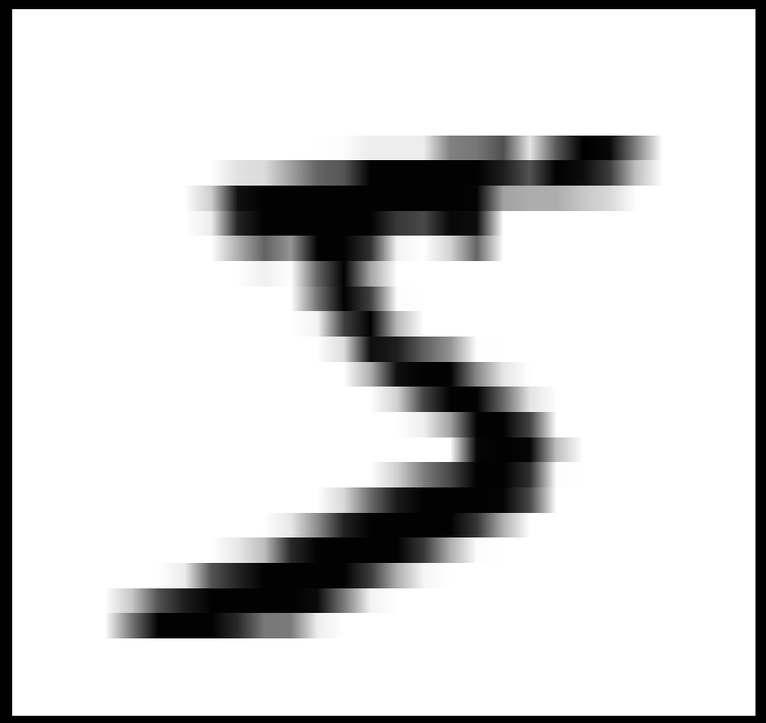

28
28
784
0
1
2
3
4
5
6
7
8
9

28



28

28

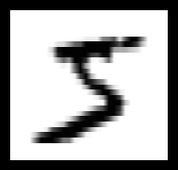


28

784

28

28



784

0

1

2

3

4

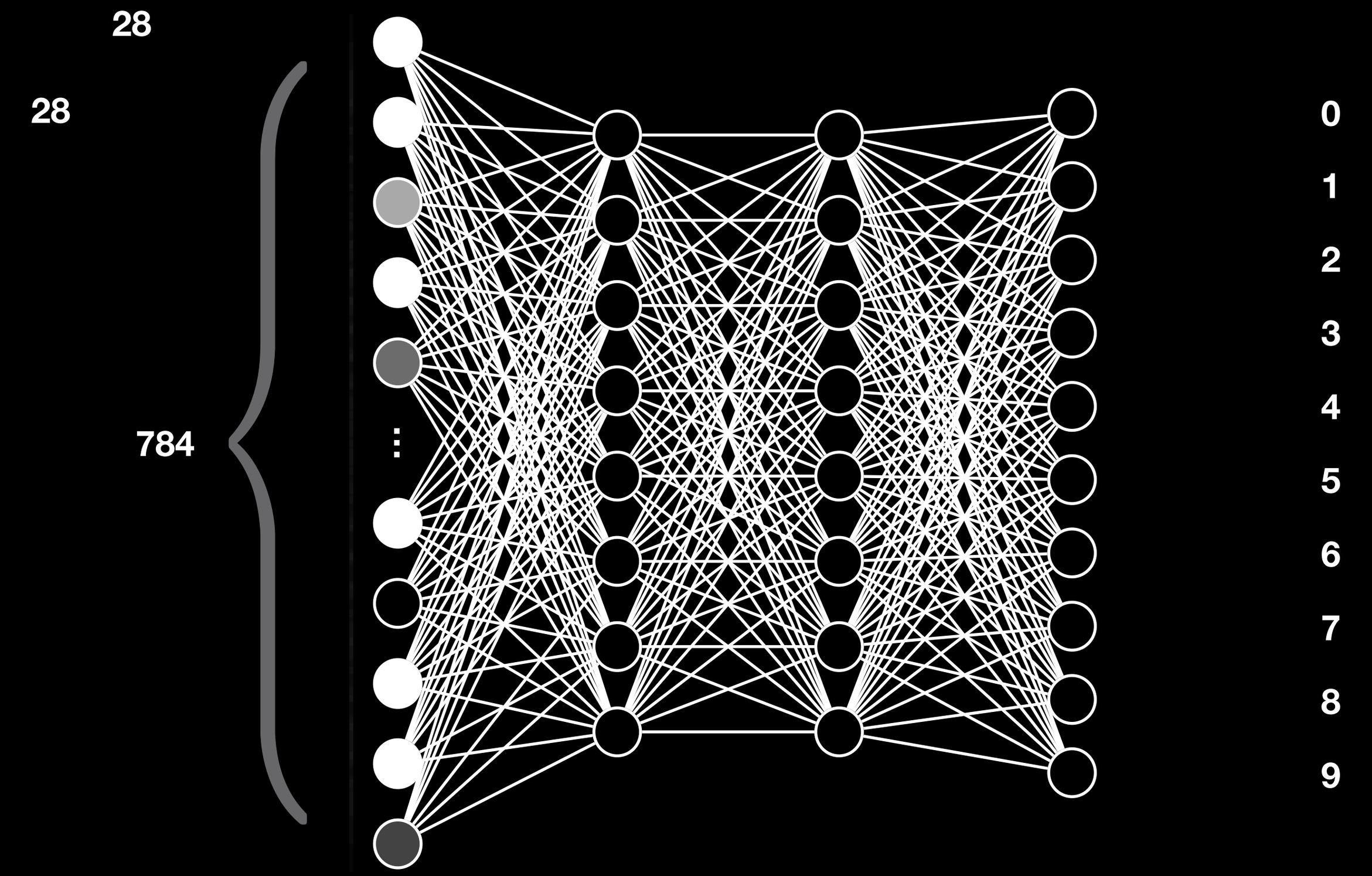
5

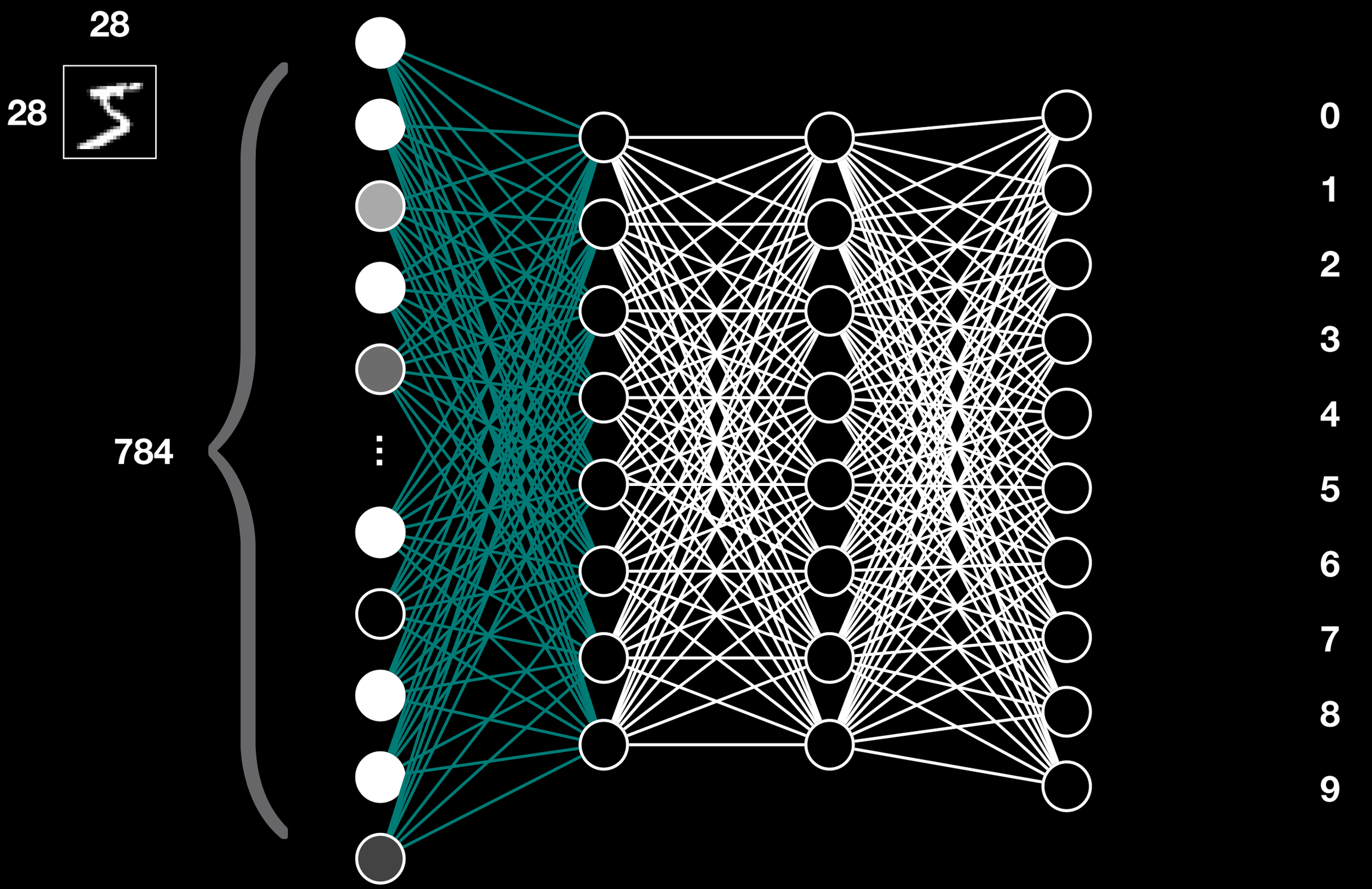
6

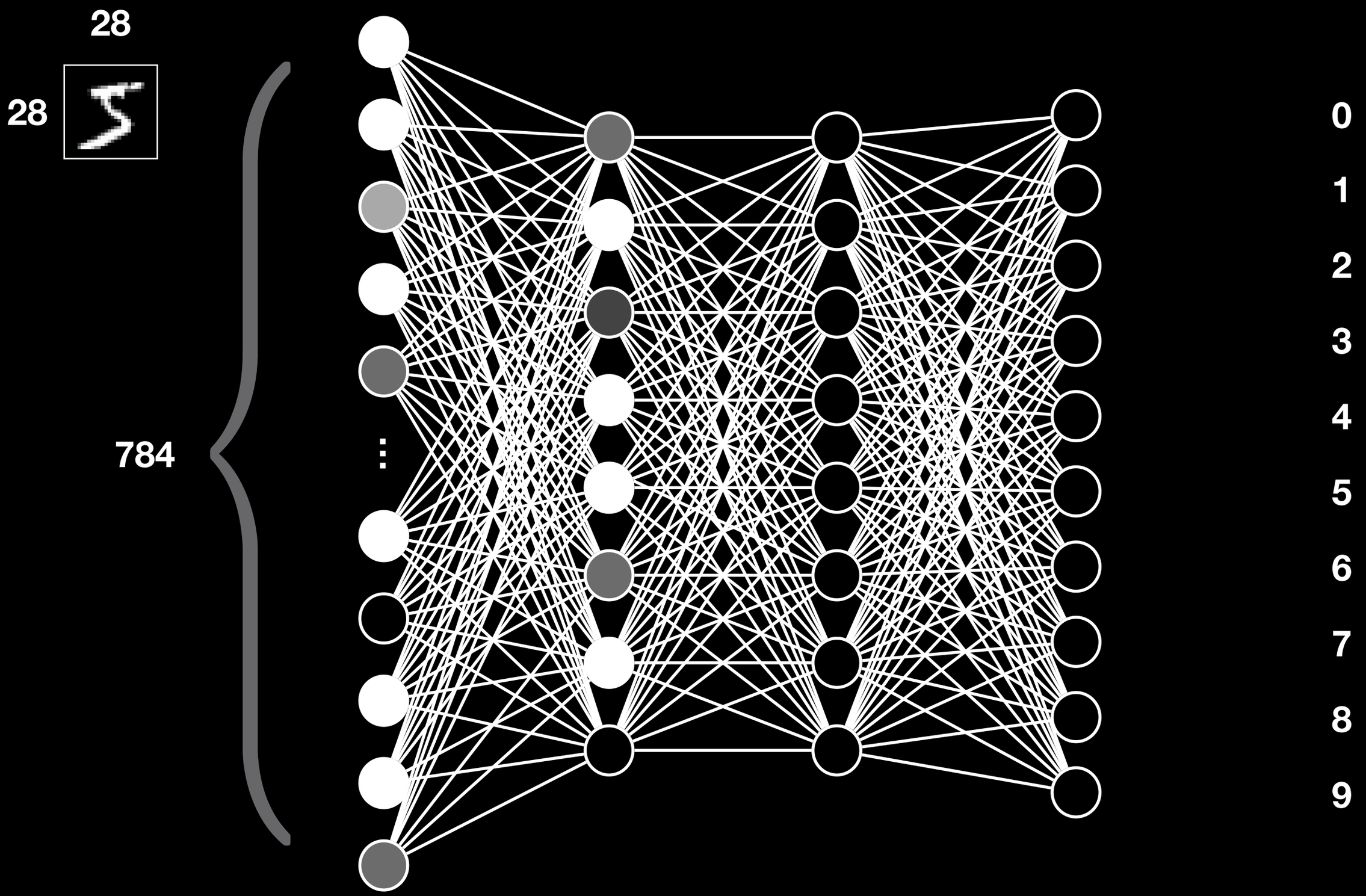
7

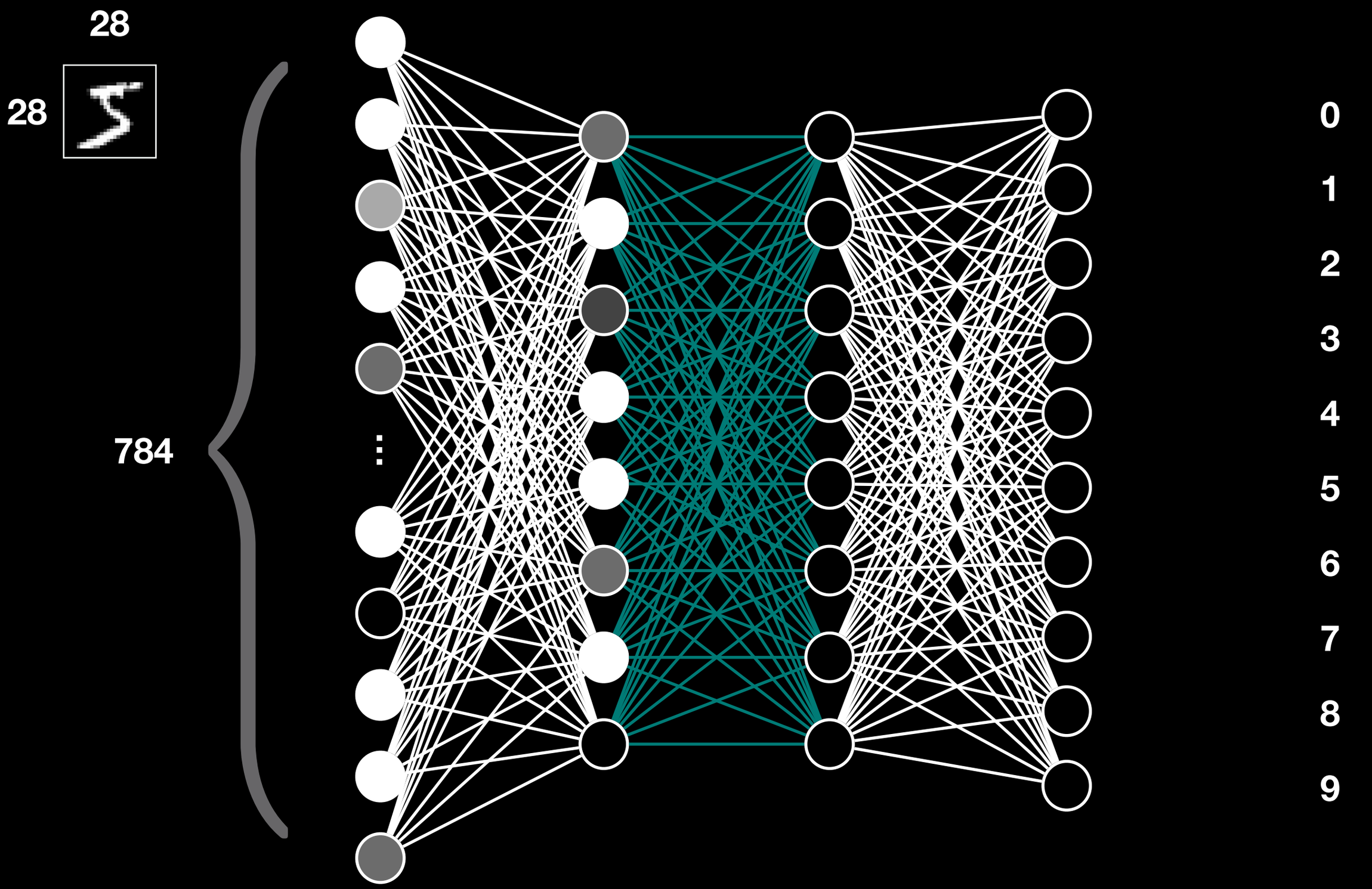
8

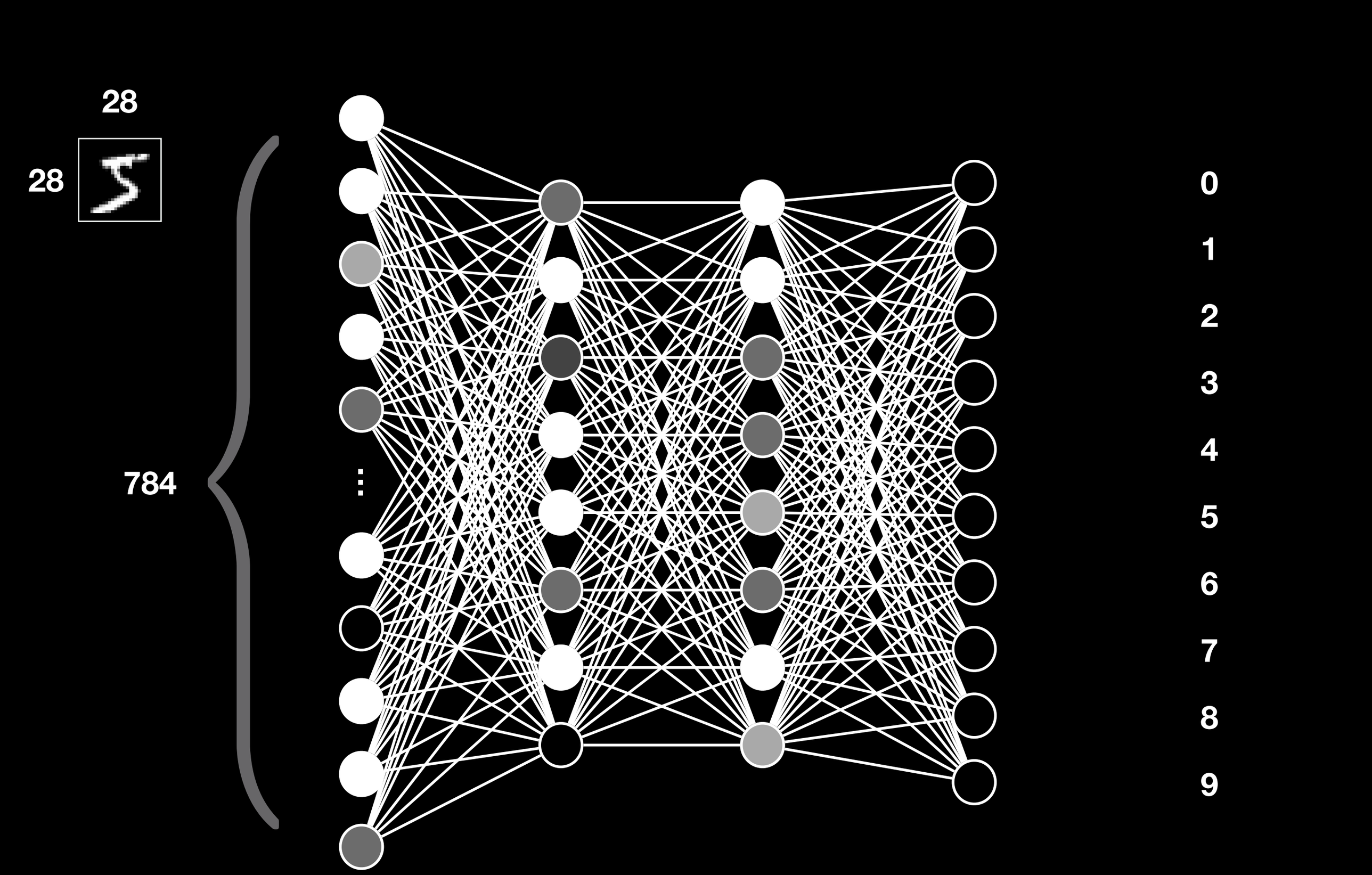
9

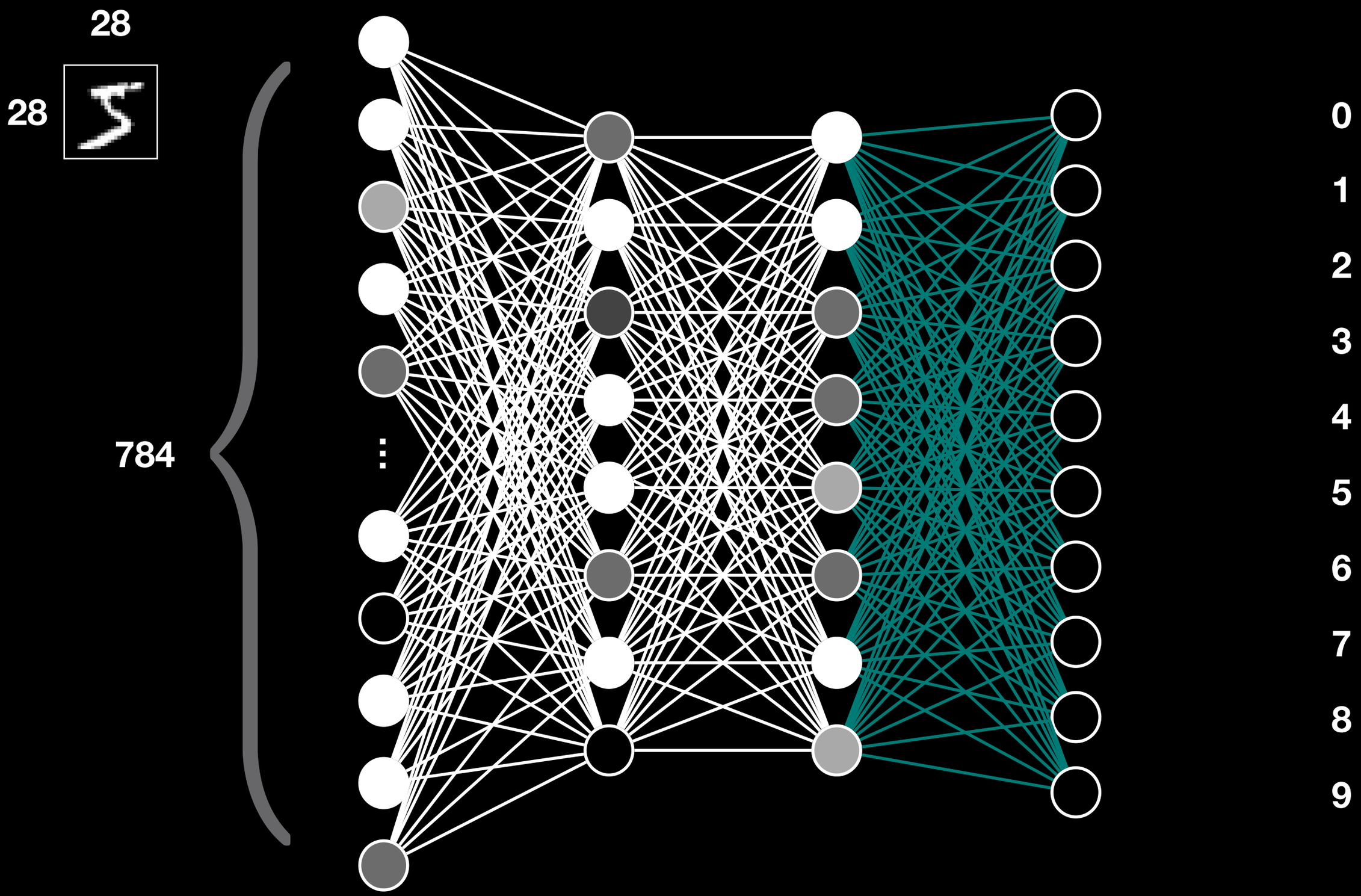

28
28
784
0
1
2
3
4
5
6
7
8
9

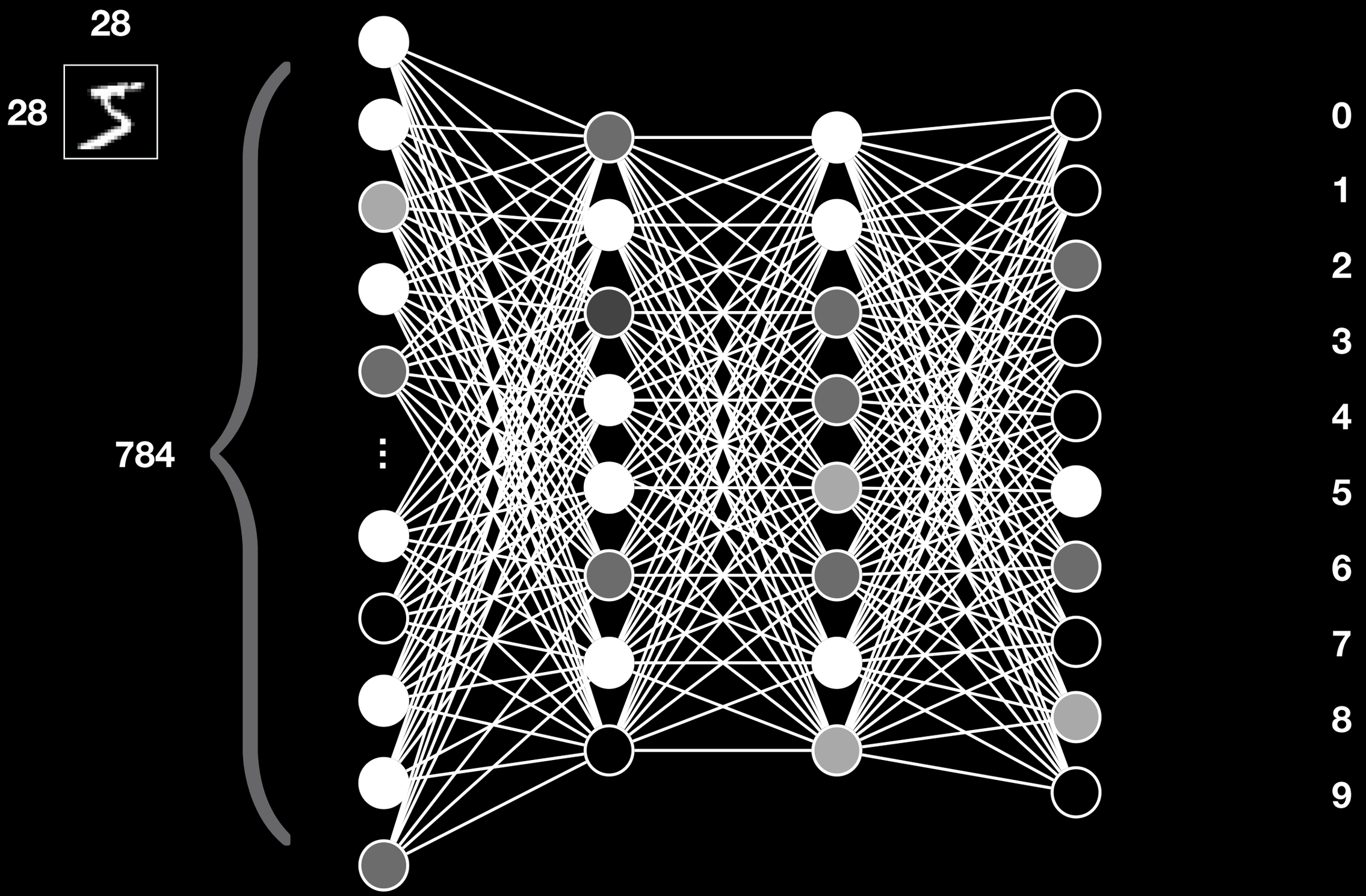

28
28
784
0
1
2
3
4
5
6
7
8
9


28
28
784
0
1
2
3
4
5
6
7
8
9


28
28
784
0
1
2
3
4
5
6
7
8
9


28
28
784
0
1
2
3
4
5
6
7
8
9


28
28
784
0
1
2
3
4
5
6
7
8
9


28
28
784
0
1
2
3
4
5
6
7
8
9

28

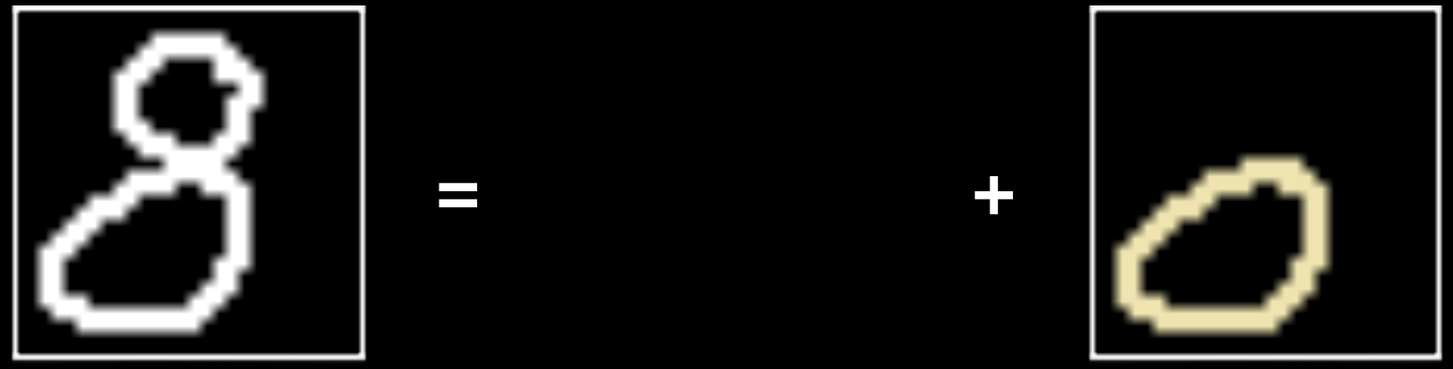
28

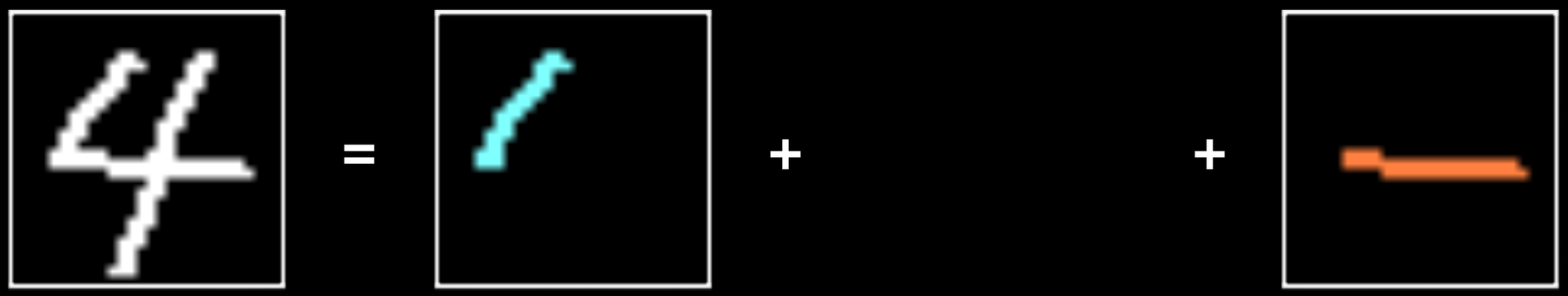
784

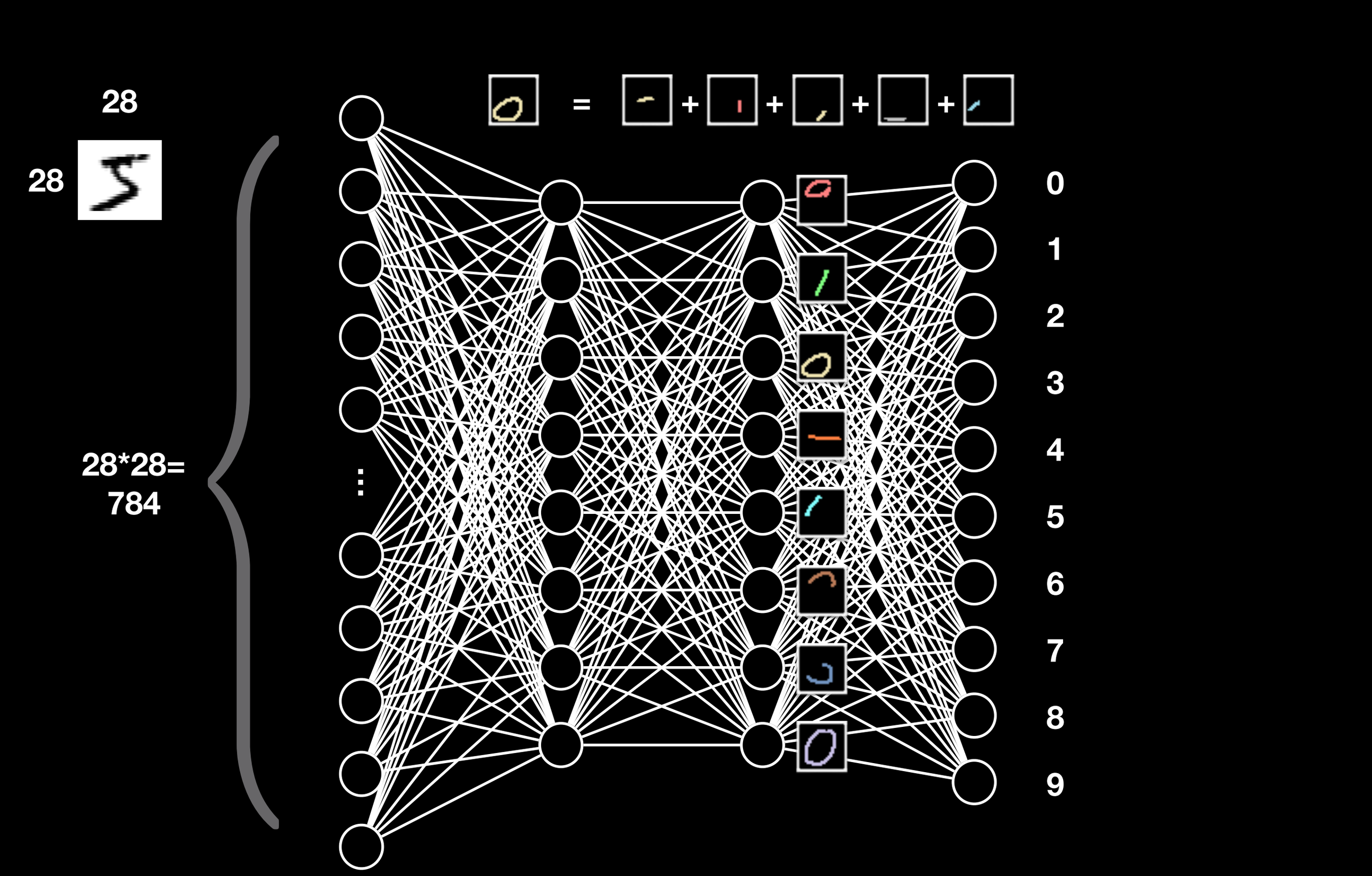
Probability

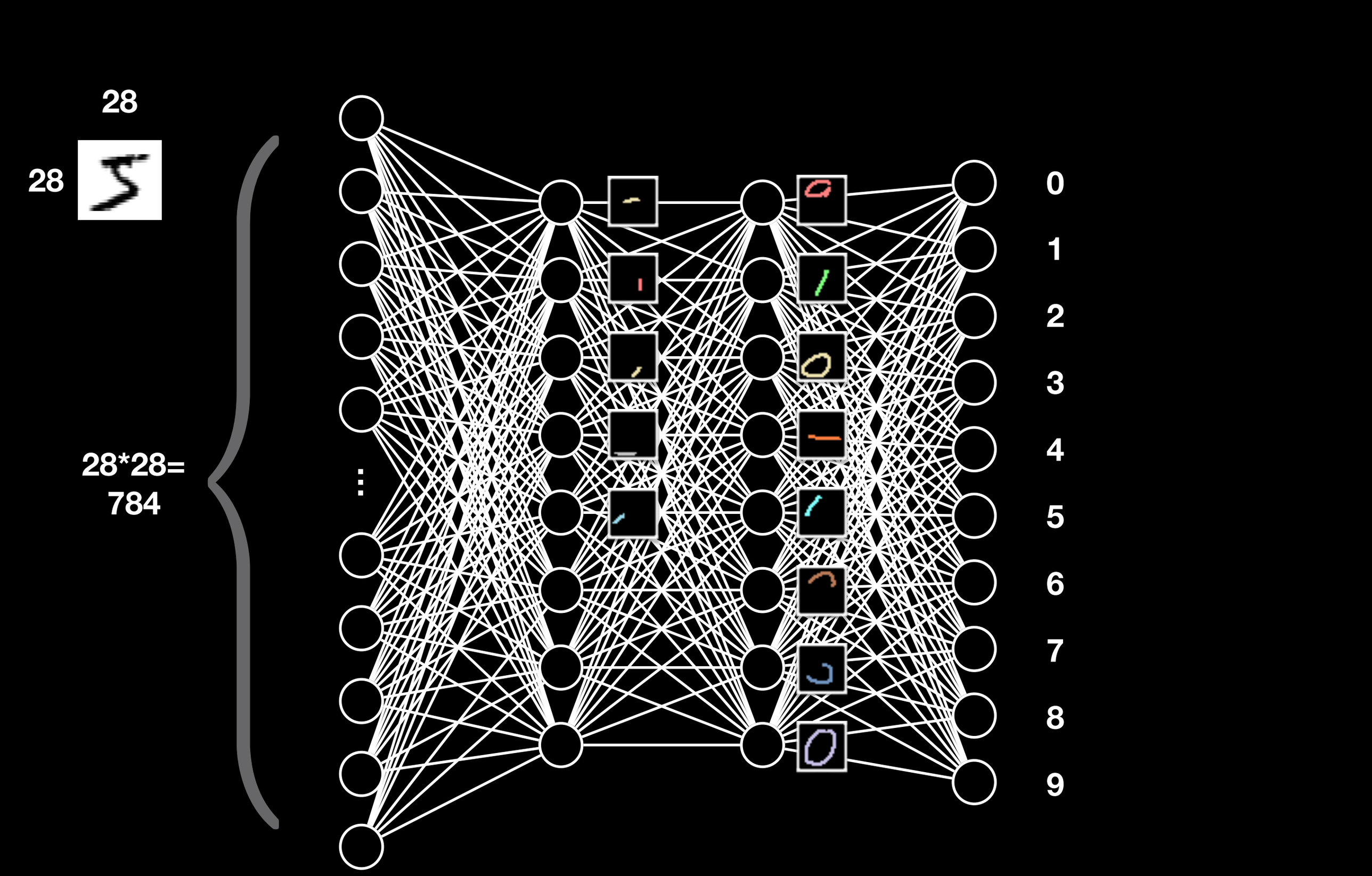
0	0
0	1
10%	2
0	3
0	4
60%	5
20%	6
0	7
10%	8
0	9

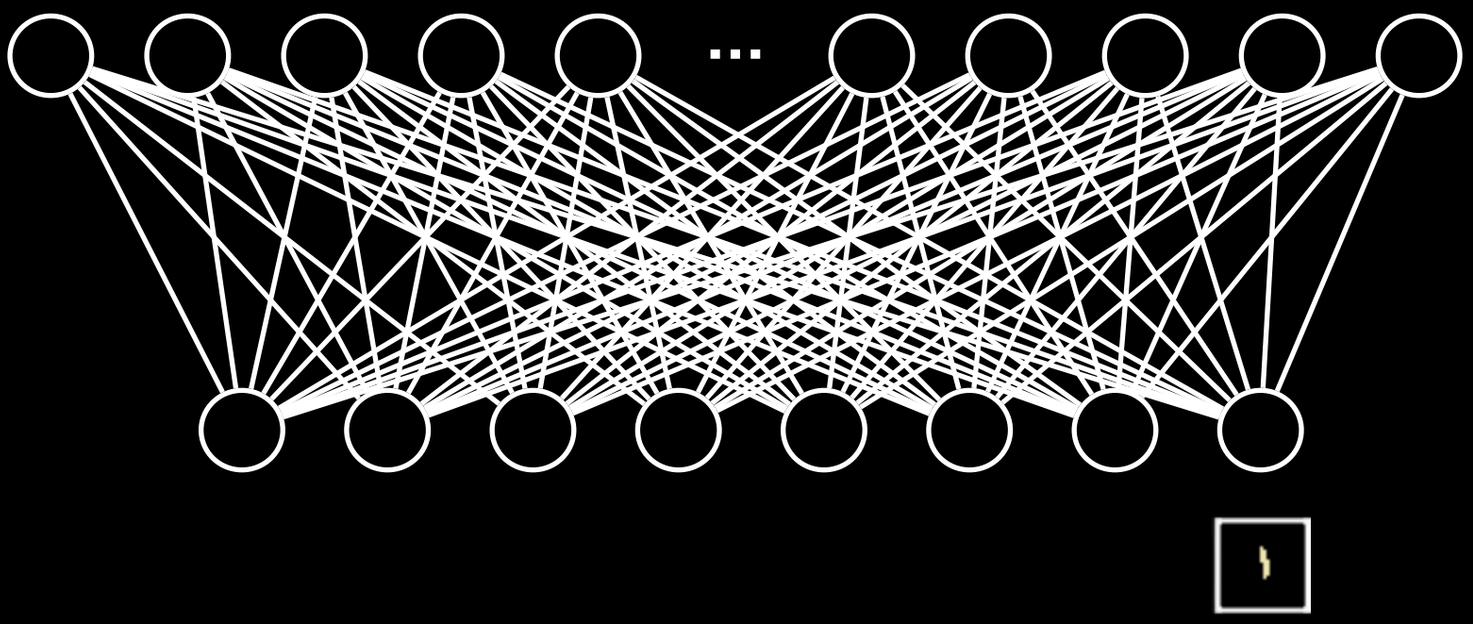
How does it work?

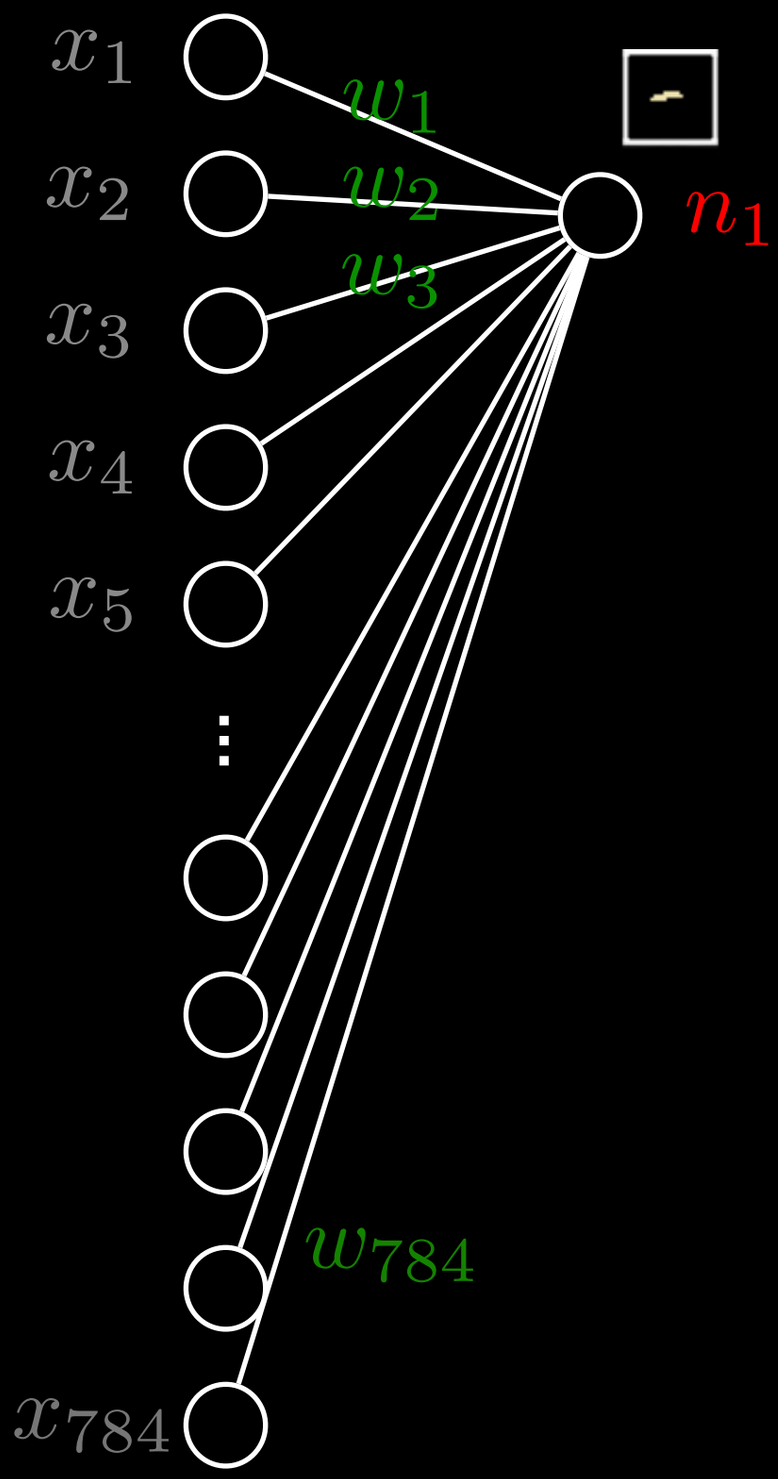





28
28
28*28=
784
=
+
+
+
+
0
1
2
3
4
5
6
7
8
9

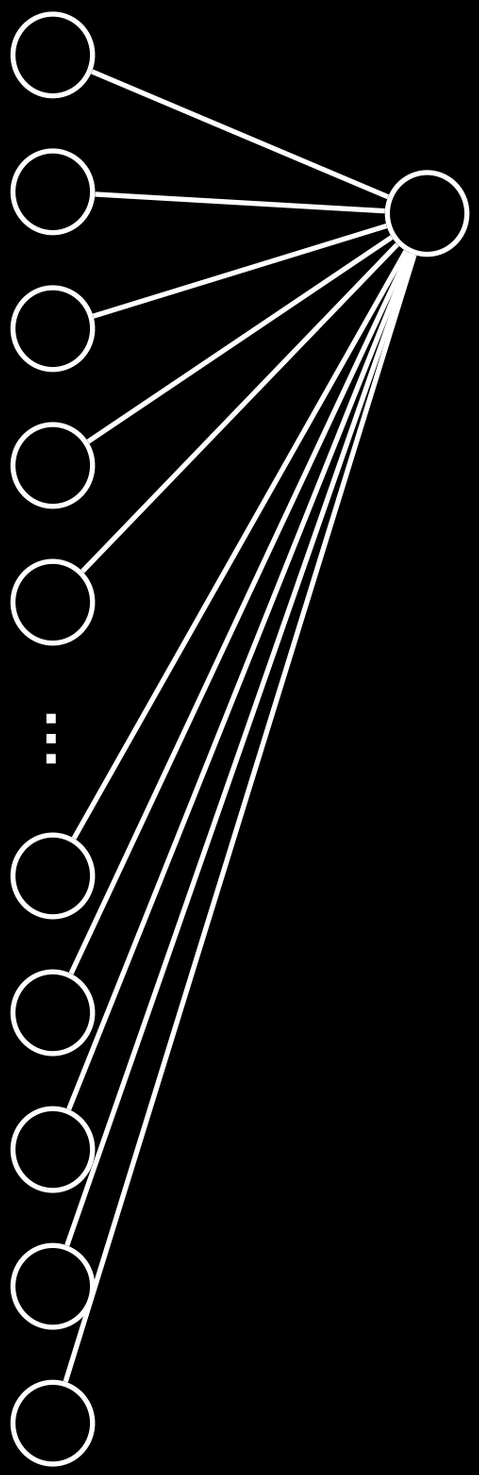

28
28
28*28=
784
0
1
2
3
4
5
6
7
8
9


...
1

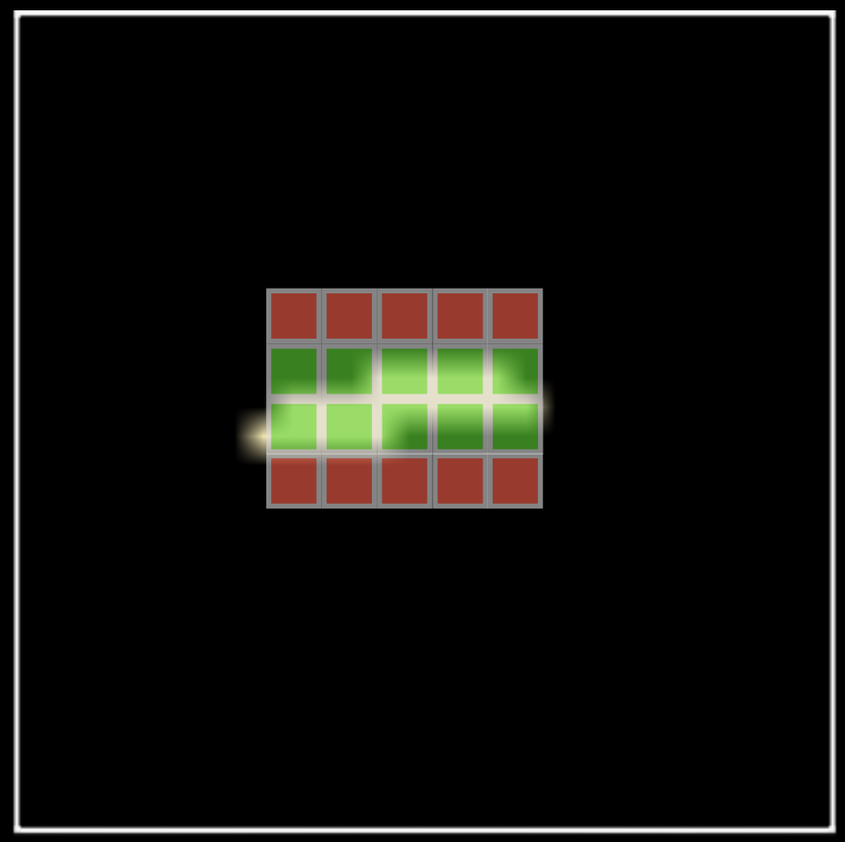


Neurons: Placeholders that take the input.

Connections: Parameters/Weights of the network.



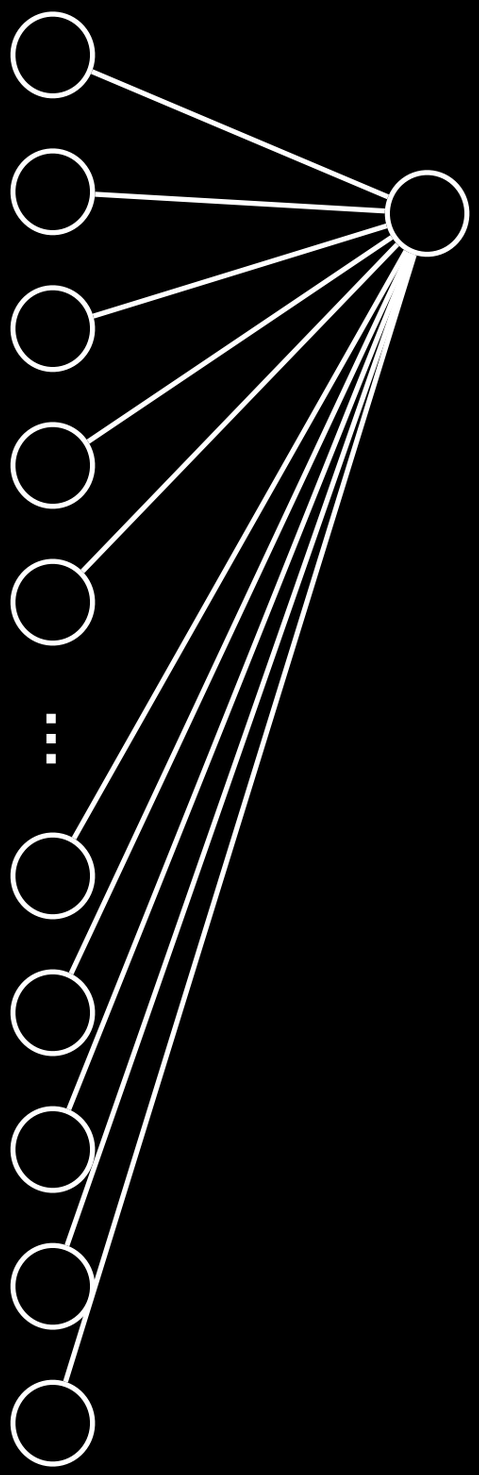
$$n_1 = (w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_{784}x_{784})$$



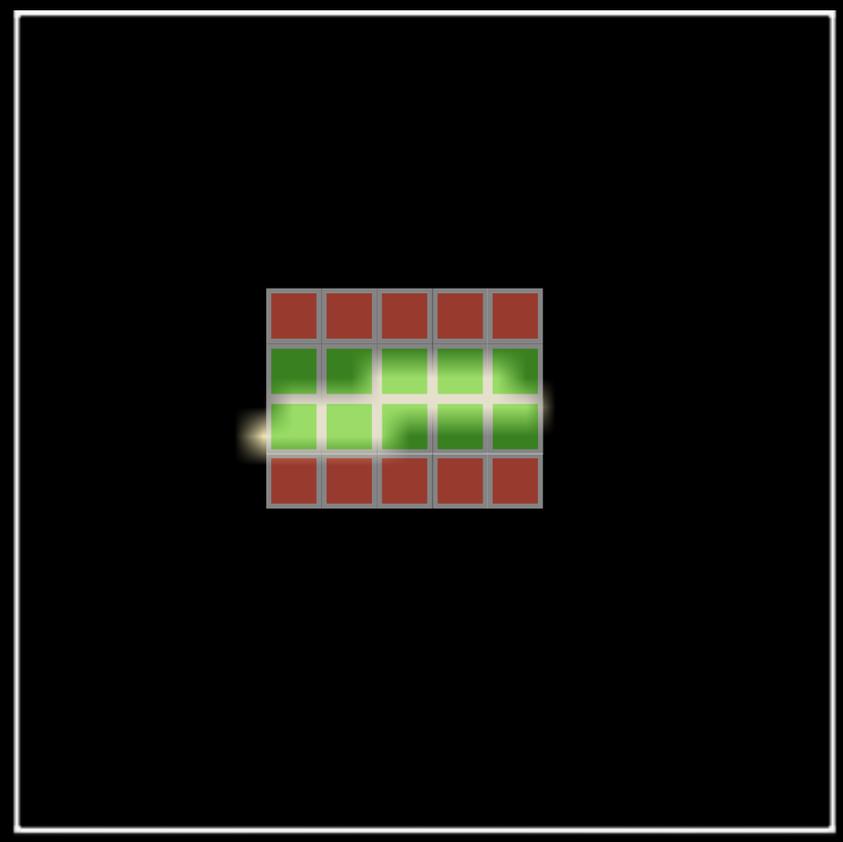
Red indicates the weights are negative
Green indicates the weights are positive

Neurons: Placeholders that take the input.

Connections: Parameters/Weights of the network.



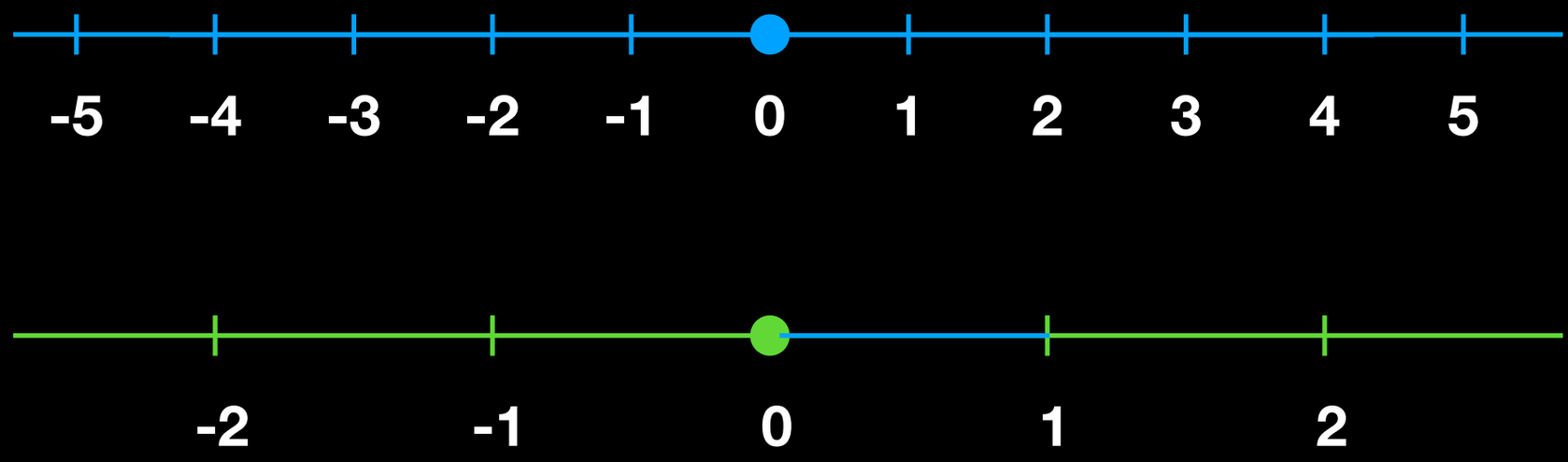
$$n_1 = (w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_{784}x_{784} + bias)$$

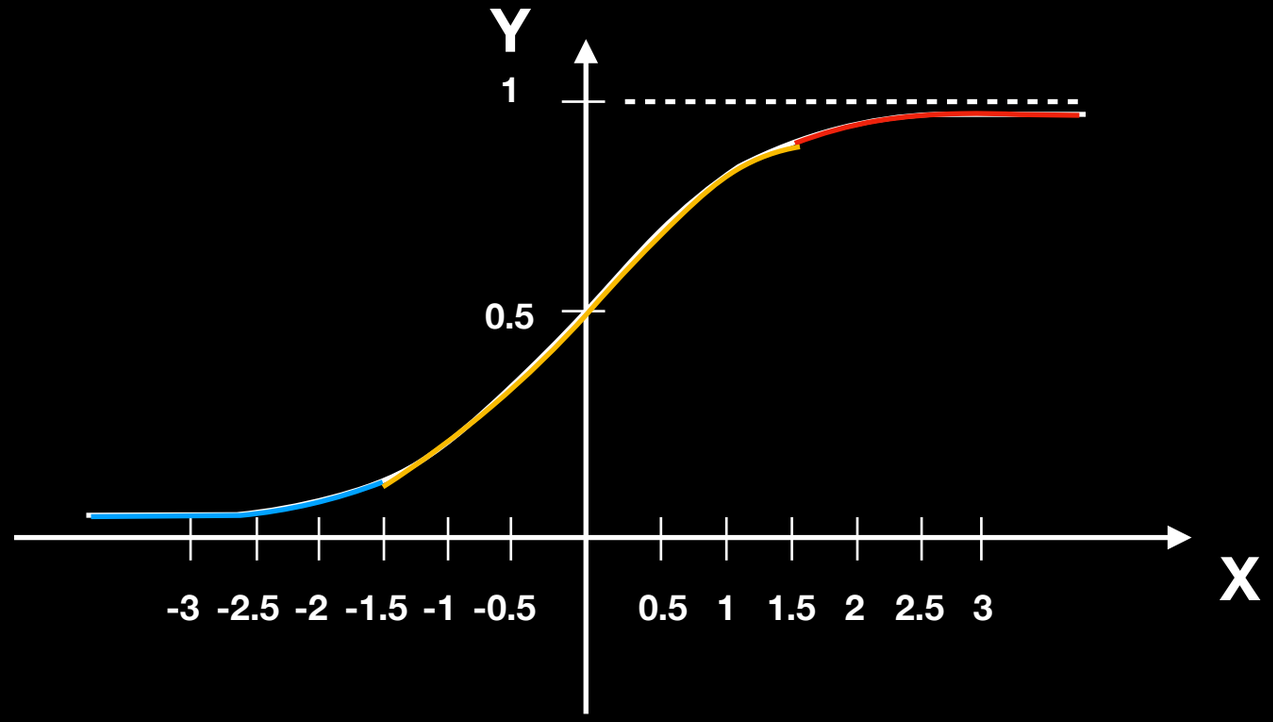


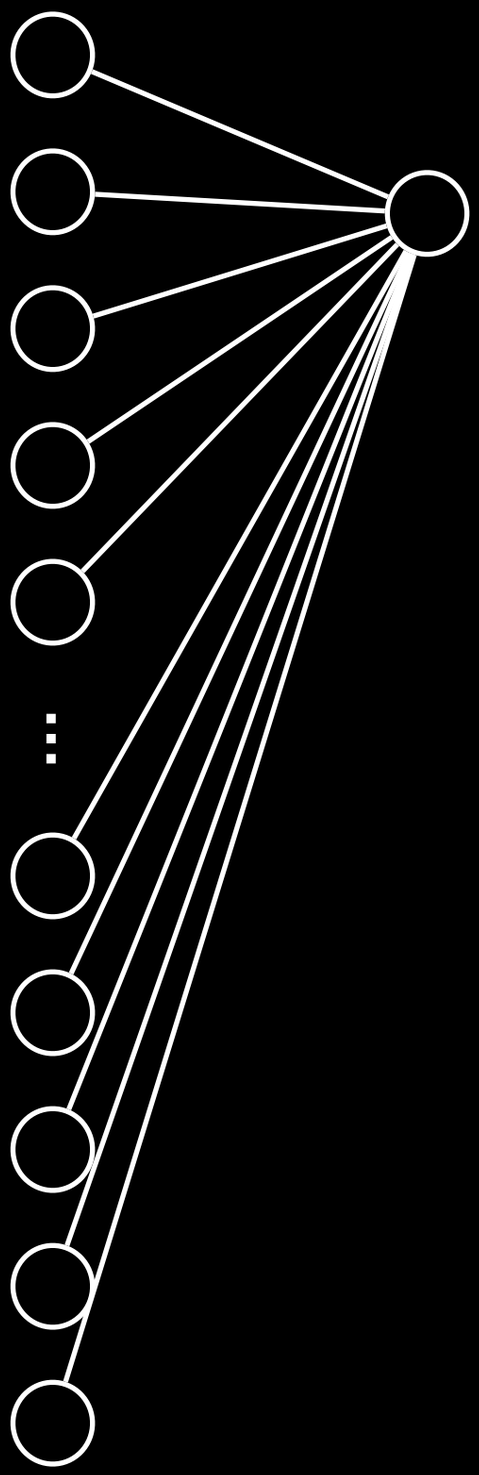
Red indicates the weights are negative
Green indicates the weights are positive

Neurone: Placeholder that takes the numbers.

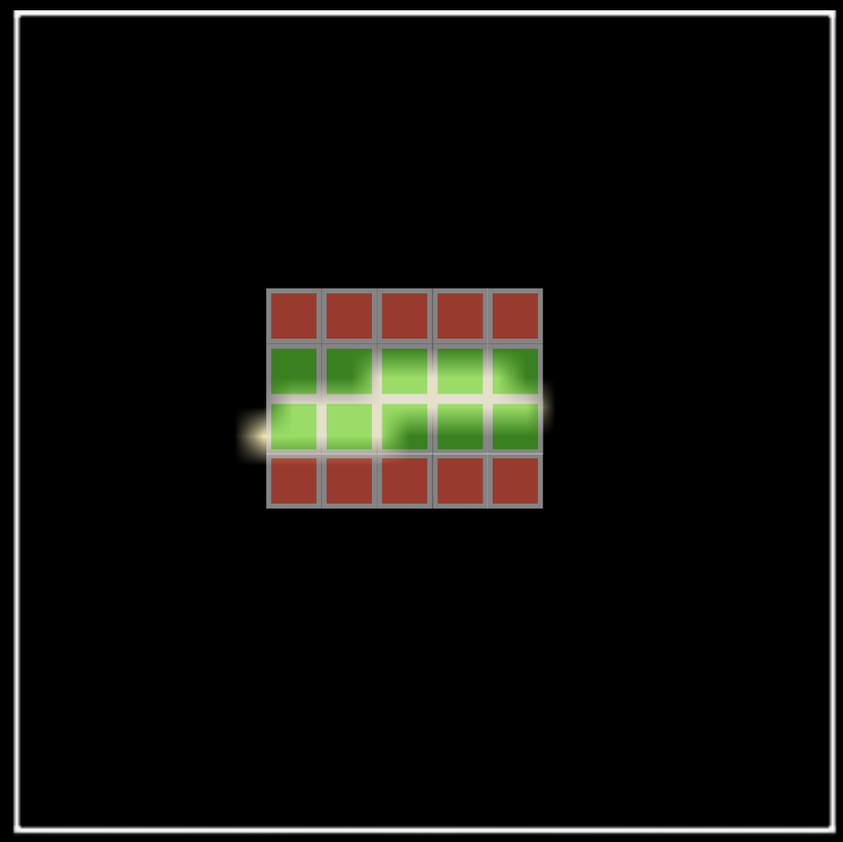
Connections: Parameters/Weights of the network.


-5
-4
-3
-2
-1
0
1
2
3
4
5
-2
-1
0
1
2


Y
1
0.5
X
-3
-2.5
-2
-1.5
-1
-0.5
0.5
1
1.5
2
2.5
3



$$n_1 = \sigma\,(w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_{784}x_{784} + bias)$$

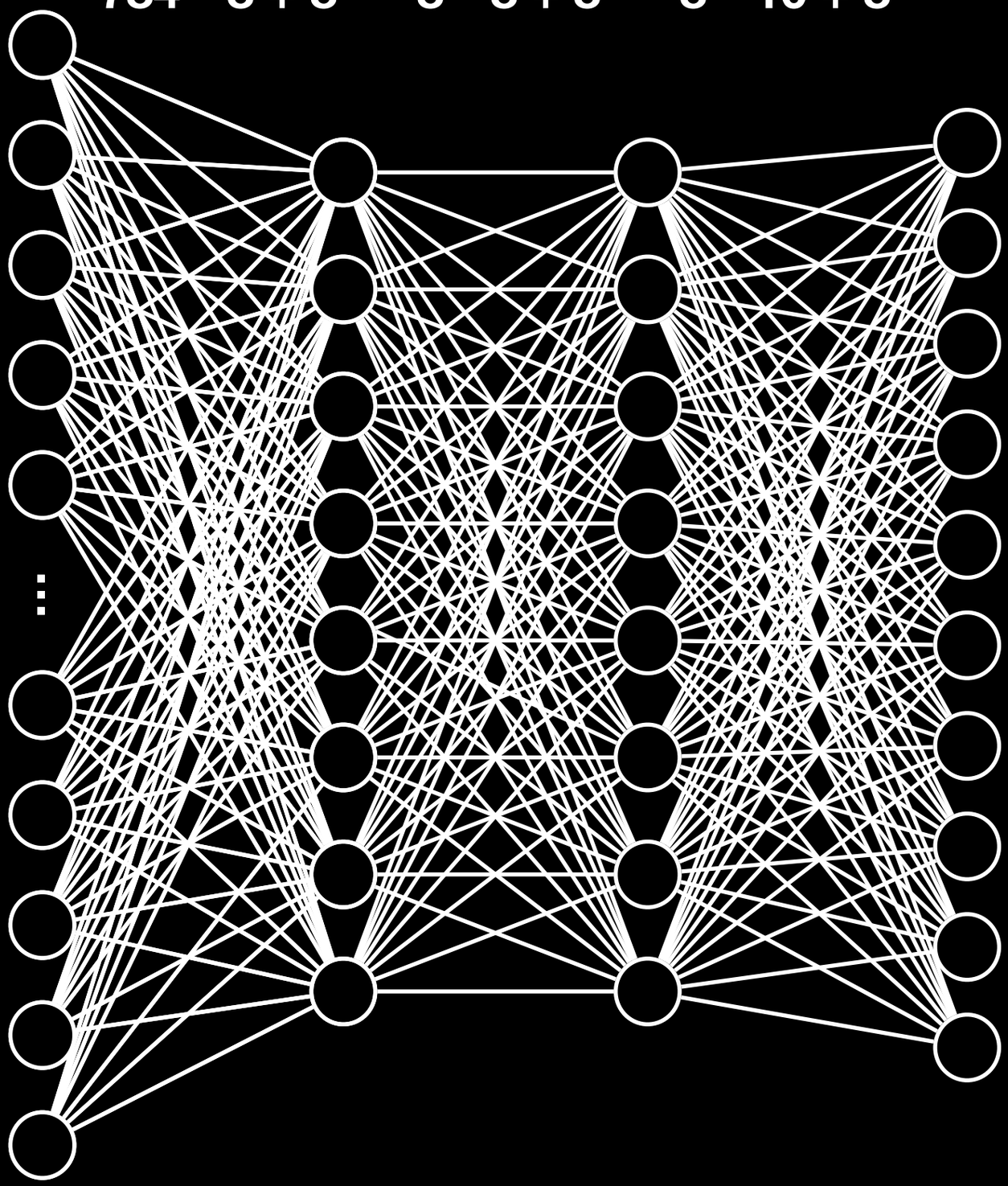


Red indicates the weights are negative
Green indicates the weights are positive

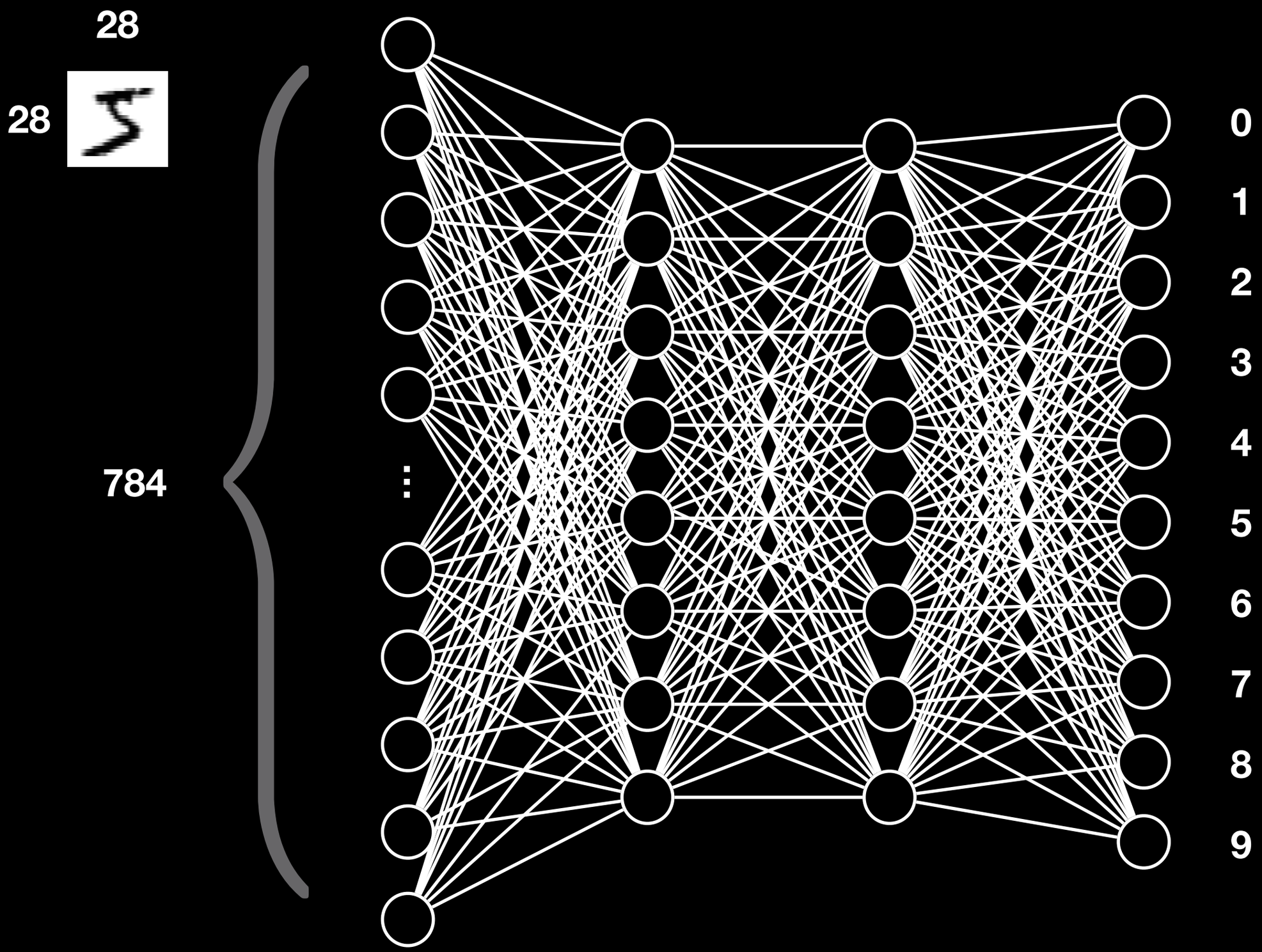
Neurons: Placeholders that take the input.

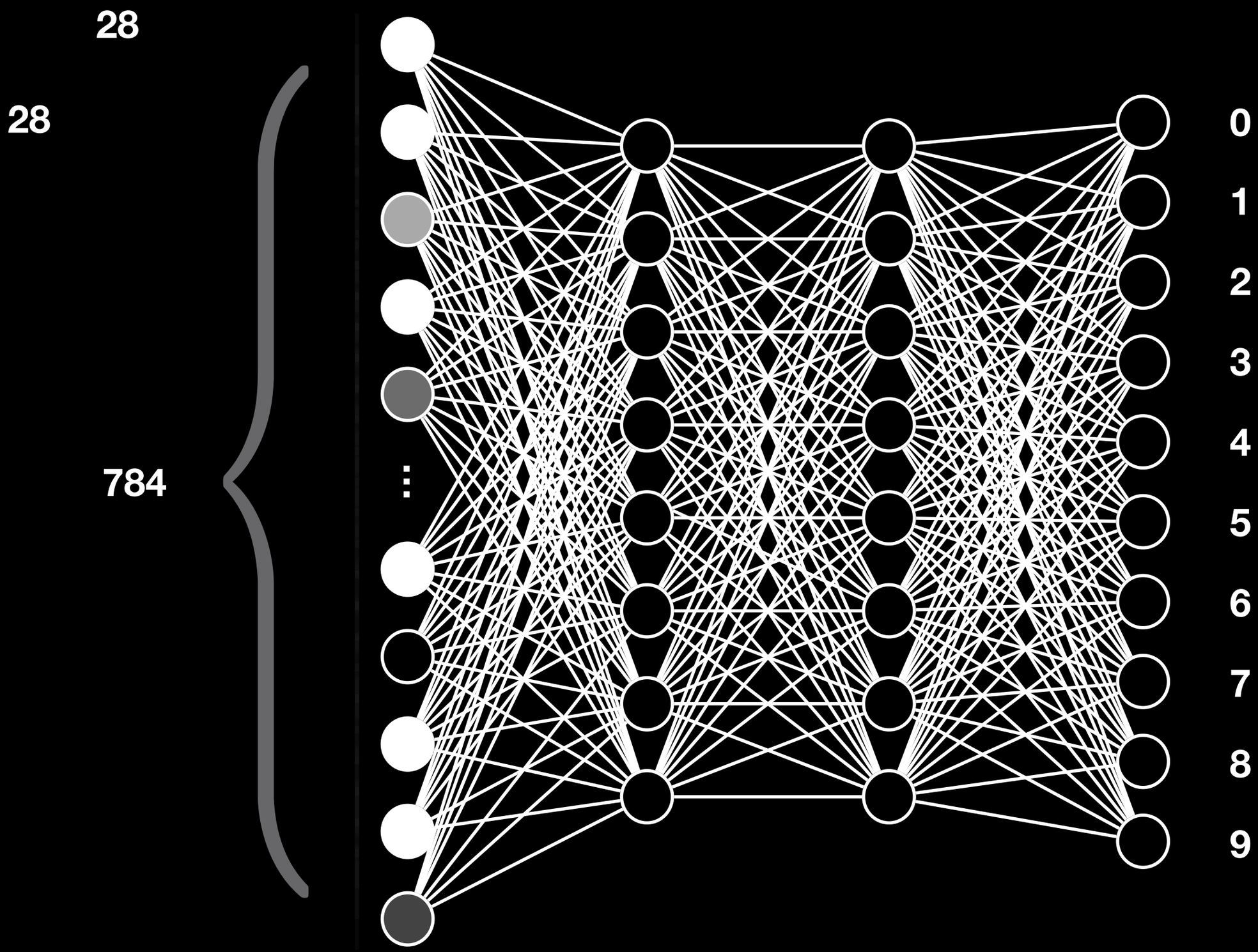
Connections: Parameters/Weights of the network.

784 * 8 + 8 8 * 8 + 8 8 * 10 + 8

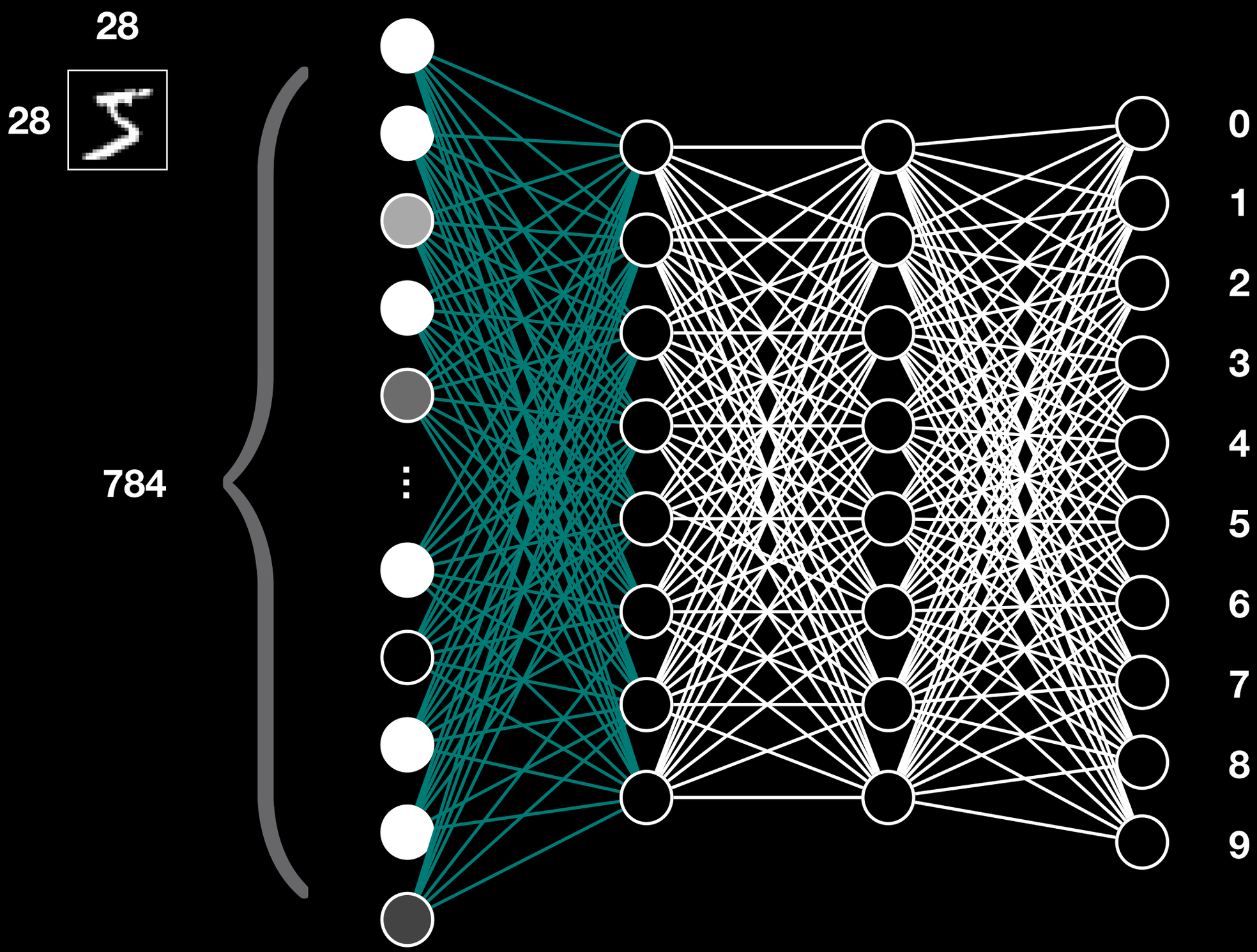


$$(c_0, c_1, ..., c_9) = f(x_1, x_2, ...x_{784})$$

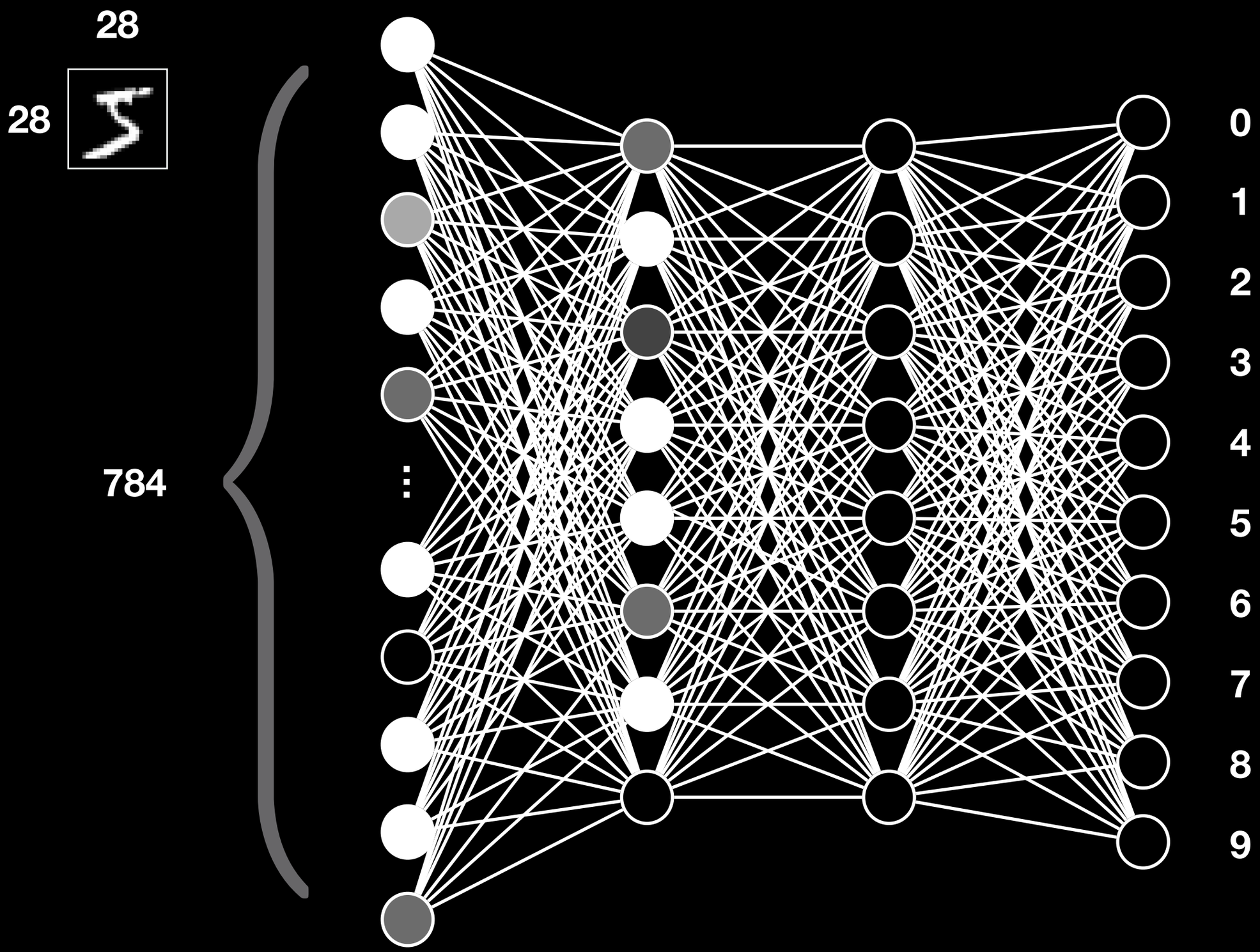

28
28
784
0
1
2
3
4
5
6
7
8
9



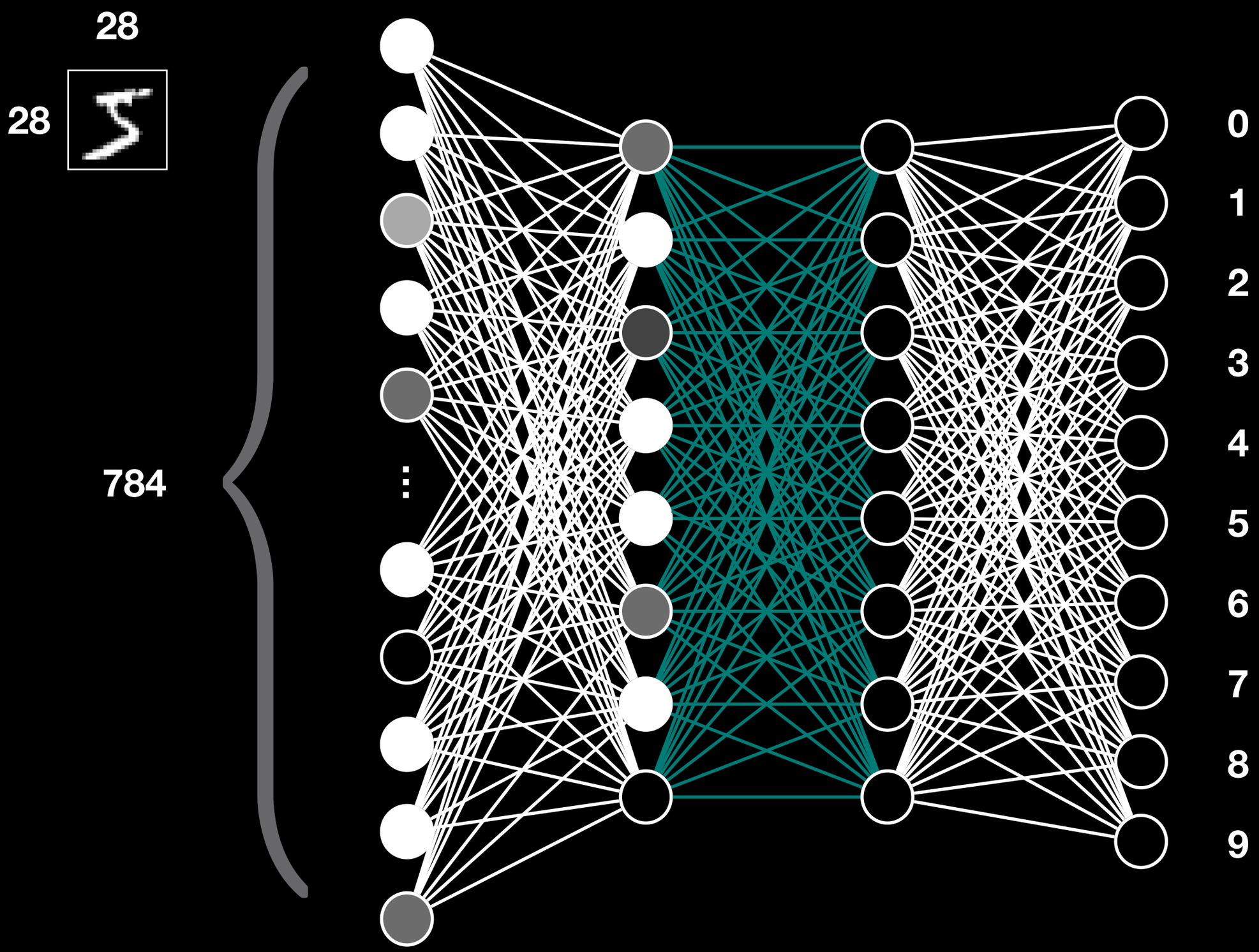
$$(c_0, c_1, ..., c_9) = f(x_1, x_2, ...x_{784})$$



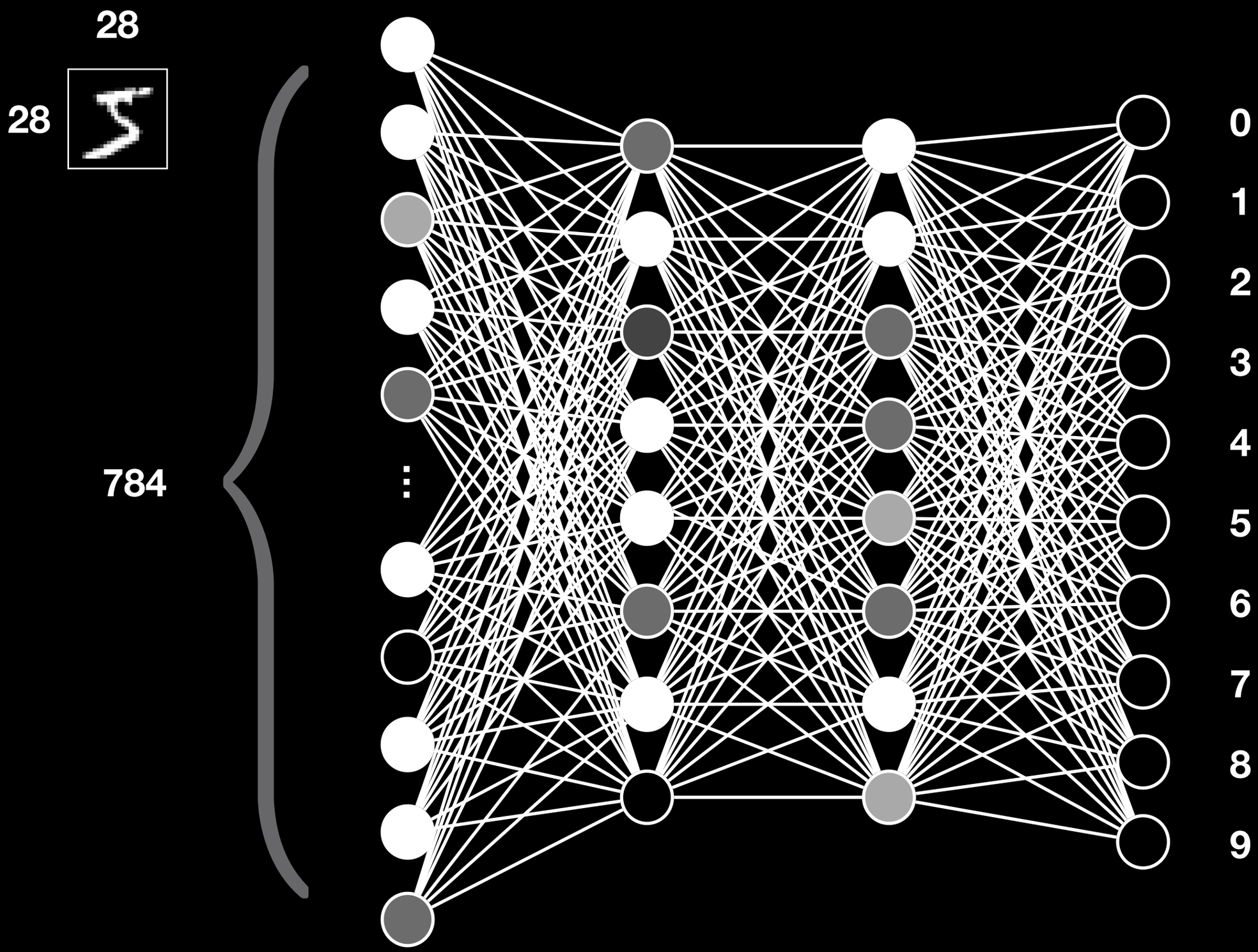
$$(c_0, c_1, ..., c_9) = f(x_1, x_2, ...x_{784})$$



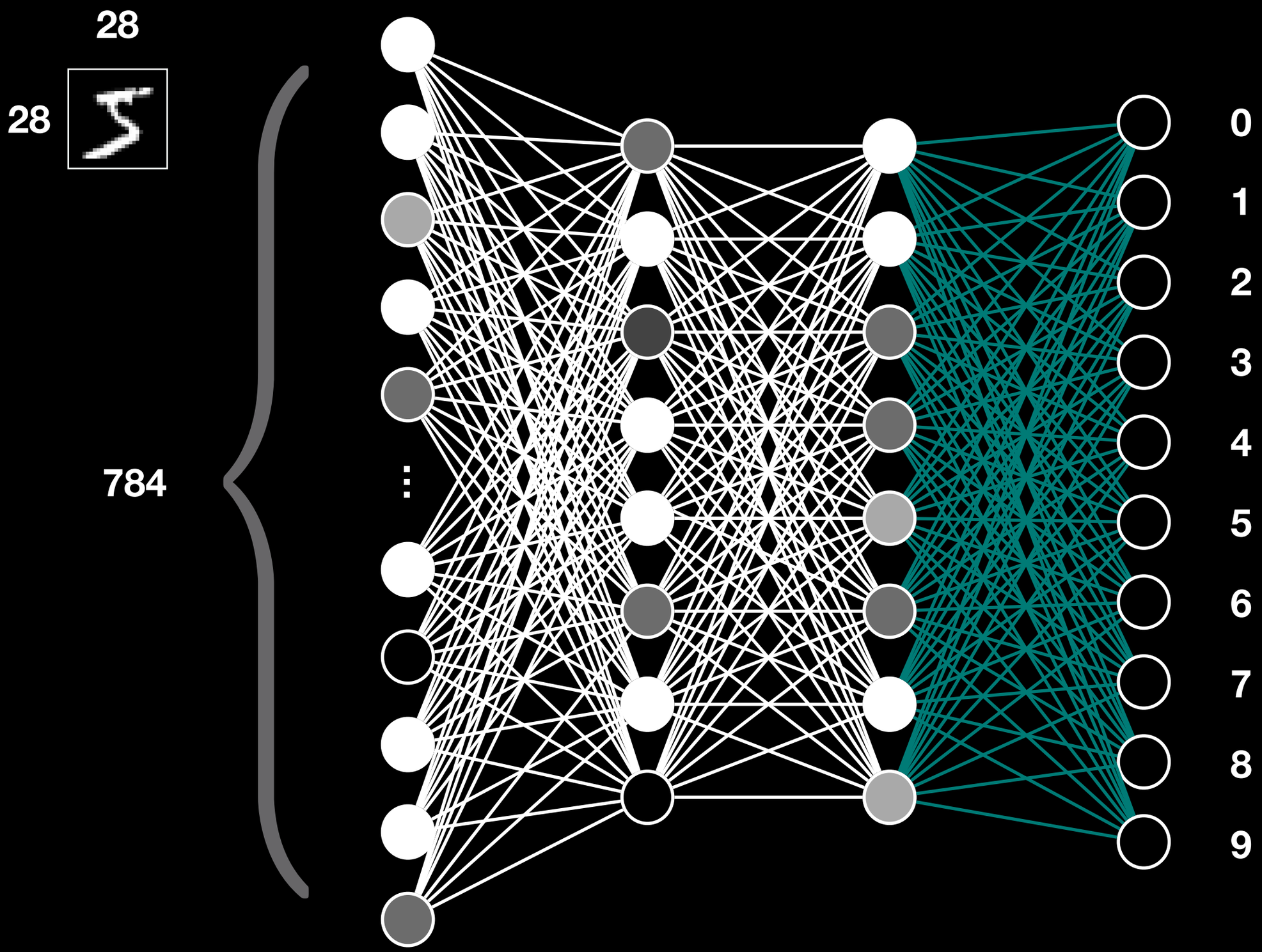
$$(c_0, c_1, ..., c_9) = f(x_1, x_2, ...x_{784})$$



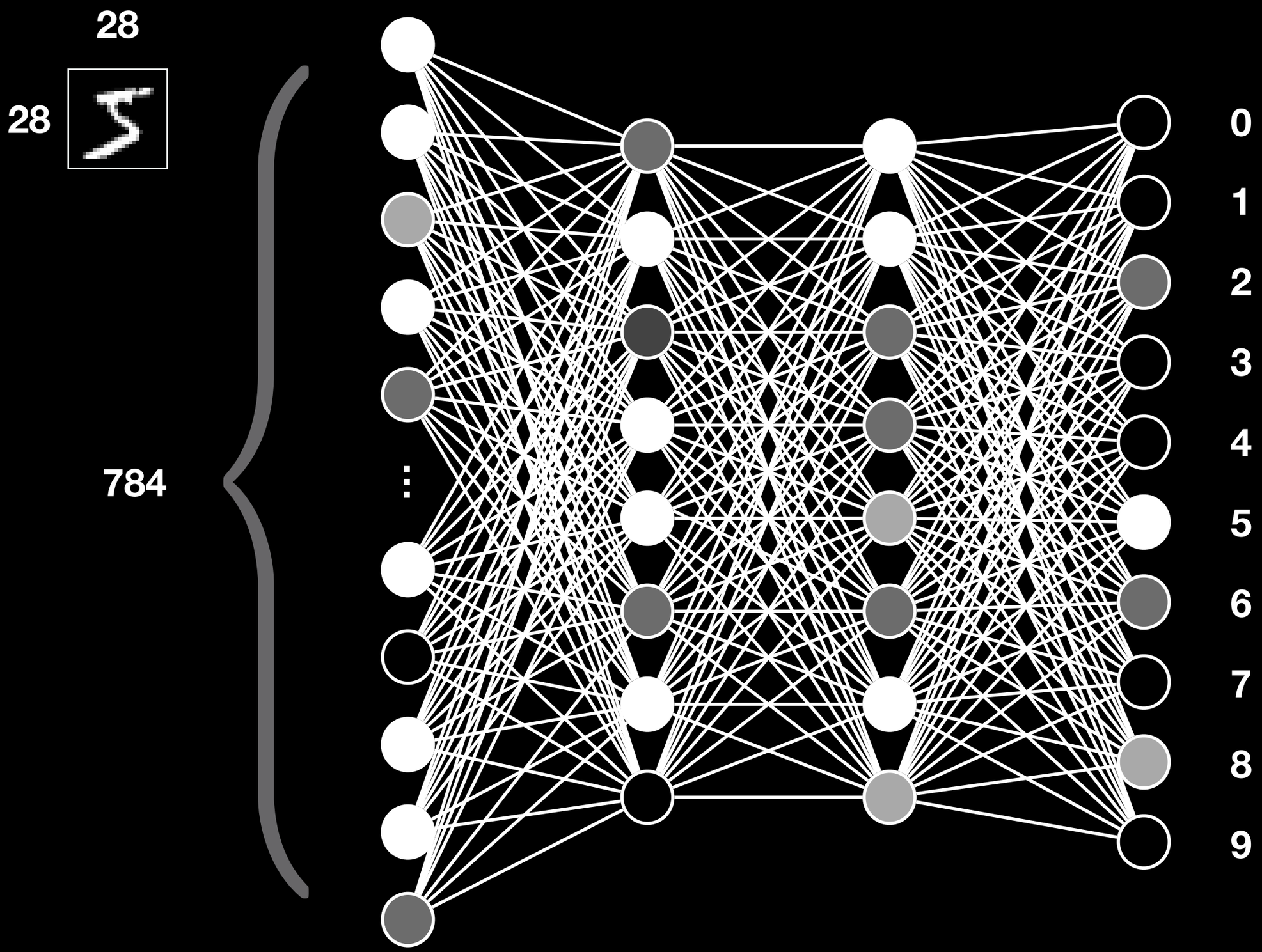
$$(c_0, c_1, ..., c_9) = f(x_1, x_2, ...x_{784})$$



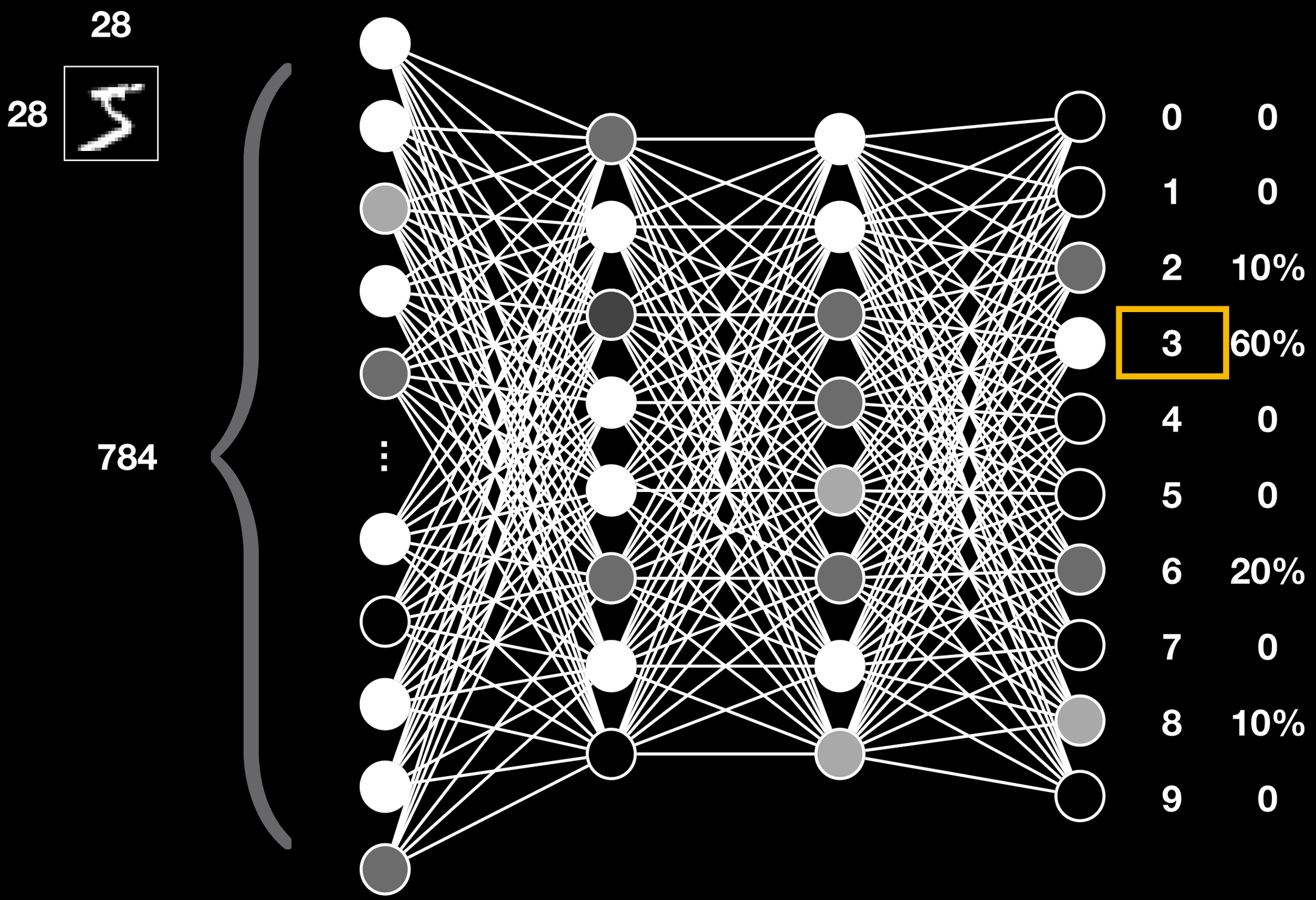
$$(c_0, c_1, ..., c_9) = f(x_1, x_2, ...x_{784})$$



$$(c_0, c_1, ..., c_9) = f(x_1, x_2, ...x_{784})$$



$$(c_0, c_1, ..., c_9) = f(x_1, x_2, ...x_{784})$$



$$(c_0, c_1, ..., c_9) = f(x_1, x_2, ...x_{784})$$

TensorFlow

Keras

theano

Install Python (Anaconda)

Install Keras (Tensorflow)

Download the dataset

Build the model and training

https://developers.google.com/machine-learning/crash-course/

bingzhang.hu@ncl.ac.uk

github.com/u112358

Caffe

PYTORCH

Caffe2