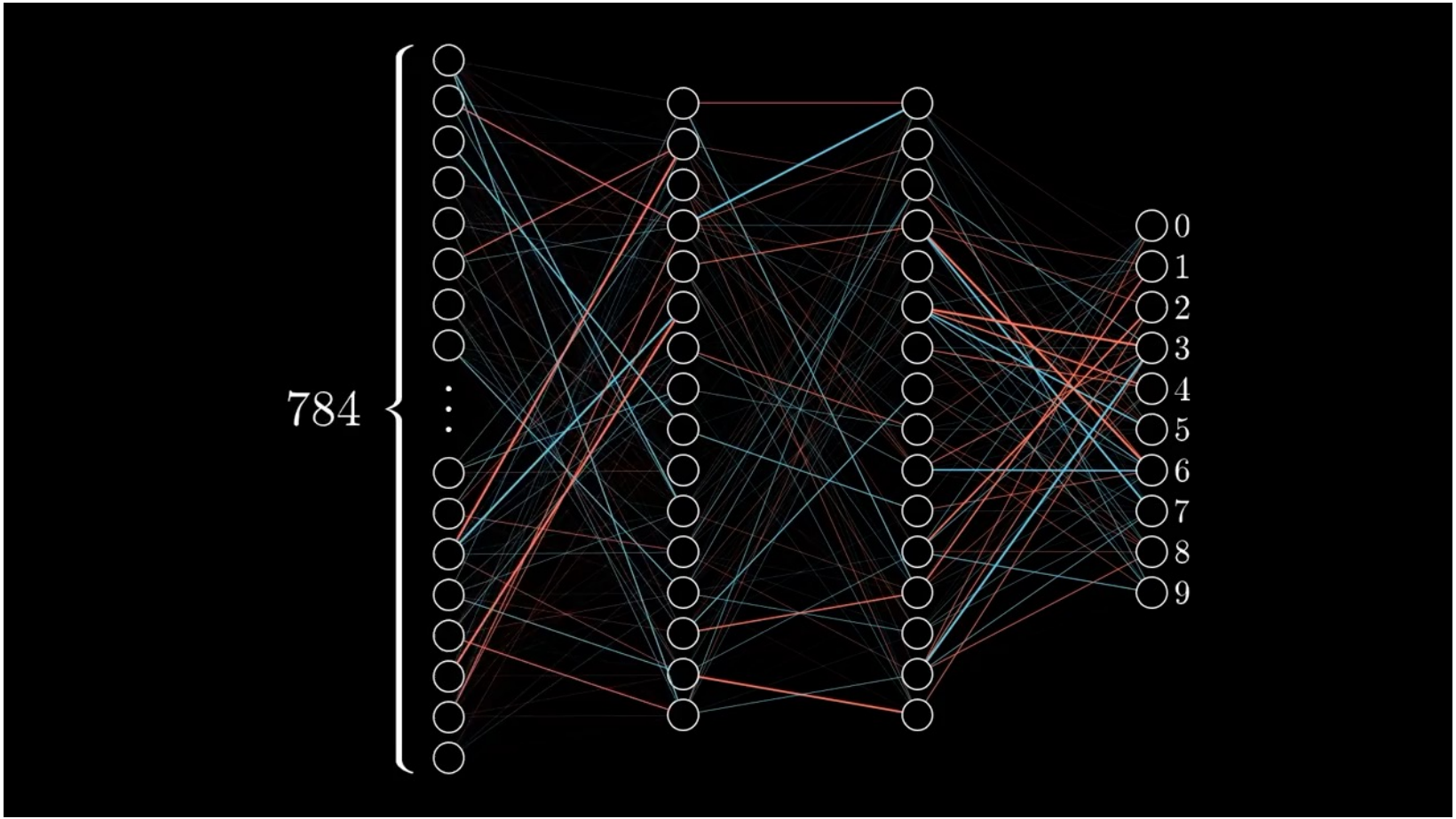
ML: DL Review Session 2

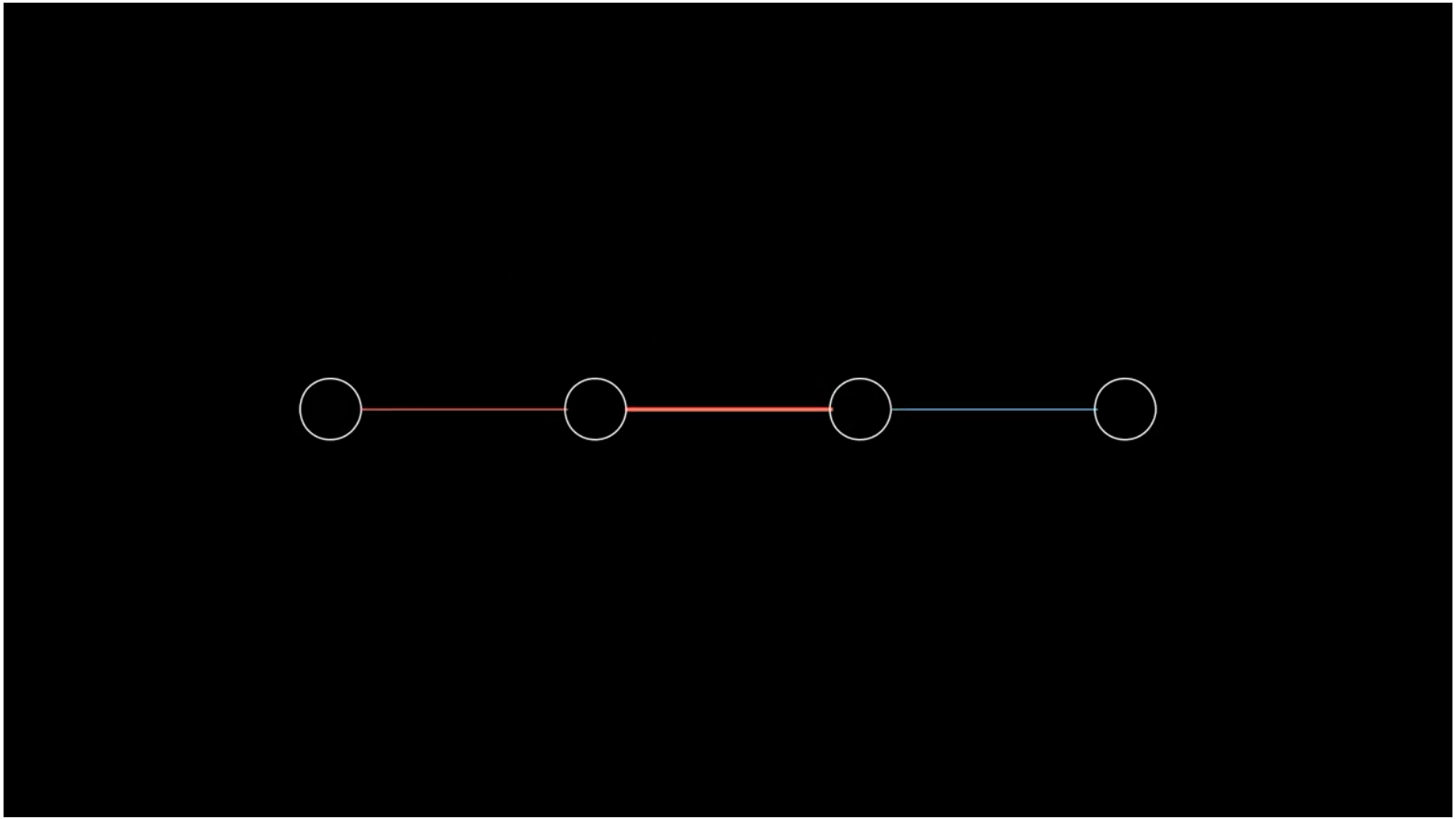
Agenda

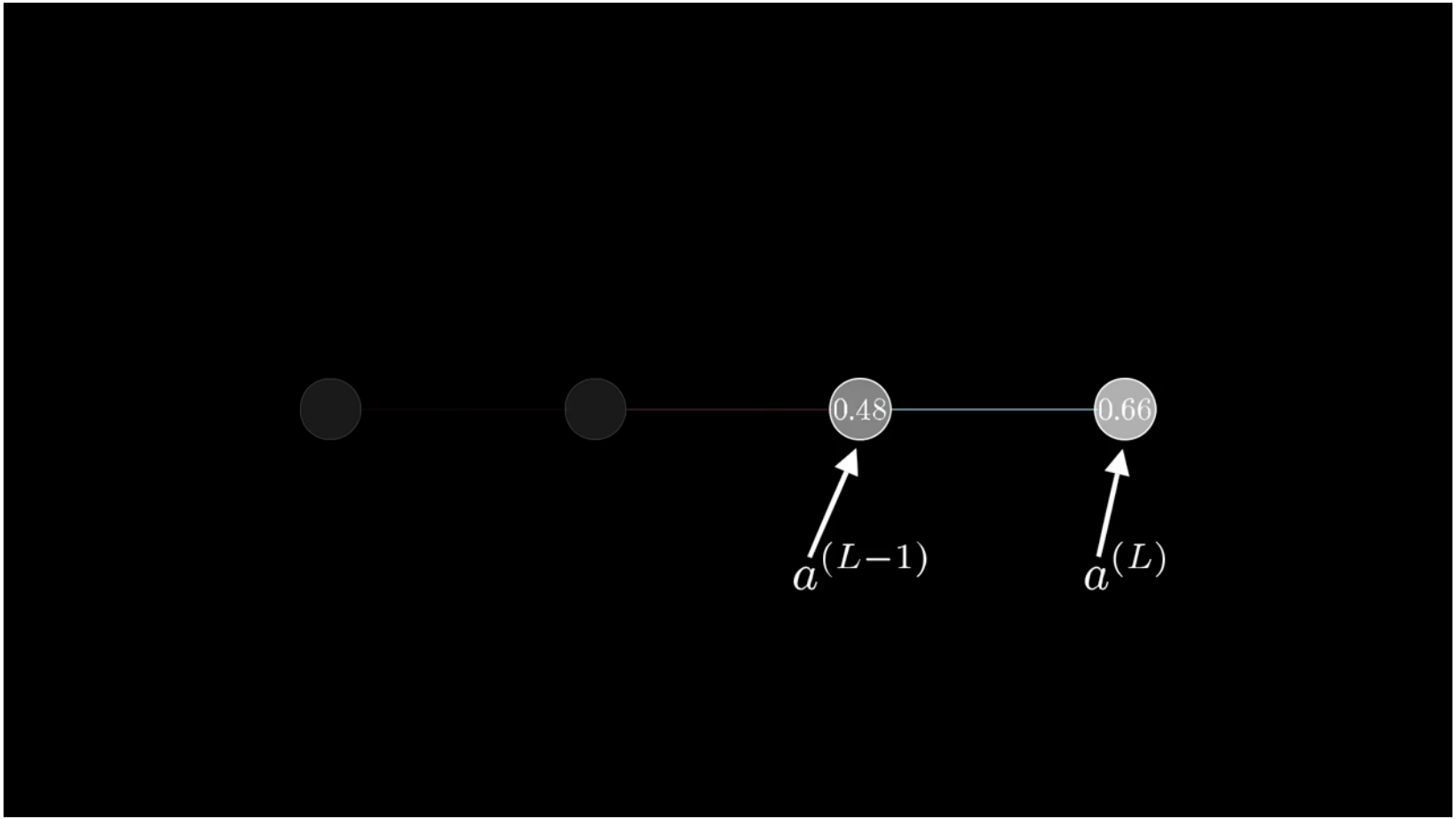
- Computational Graph / Back-prop
    - Visualization for understanding
    - Practice
- Intro to NNs
    - Activation function
    - MLP walkthrough with google colab

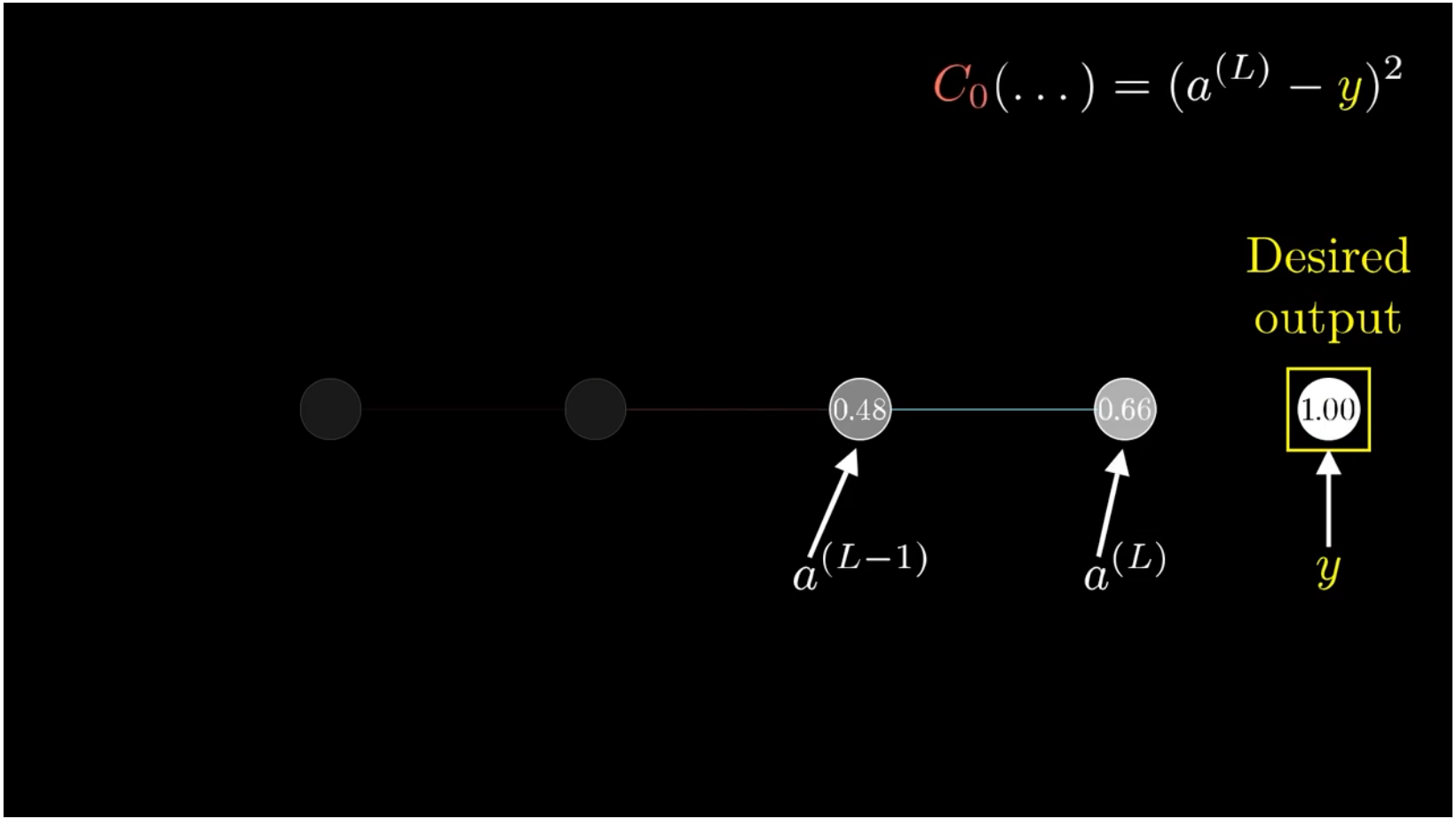
Computational graph of a Neural Net

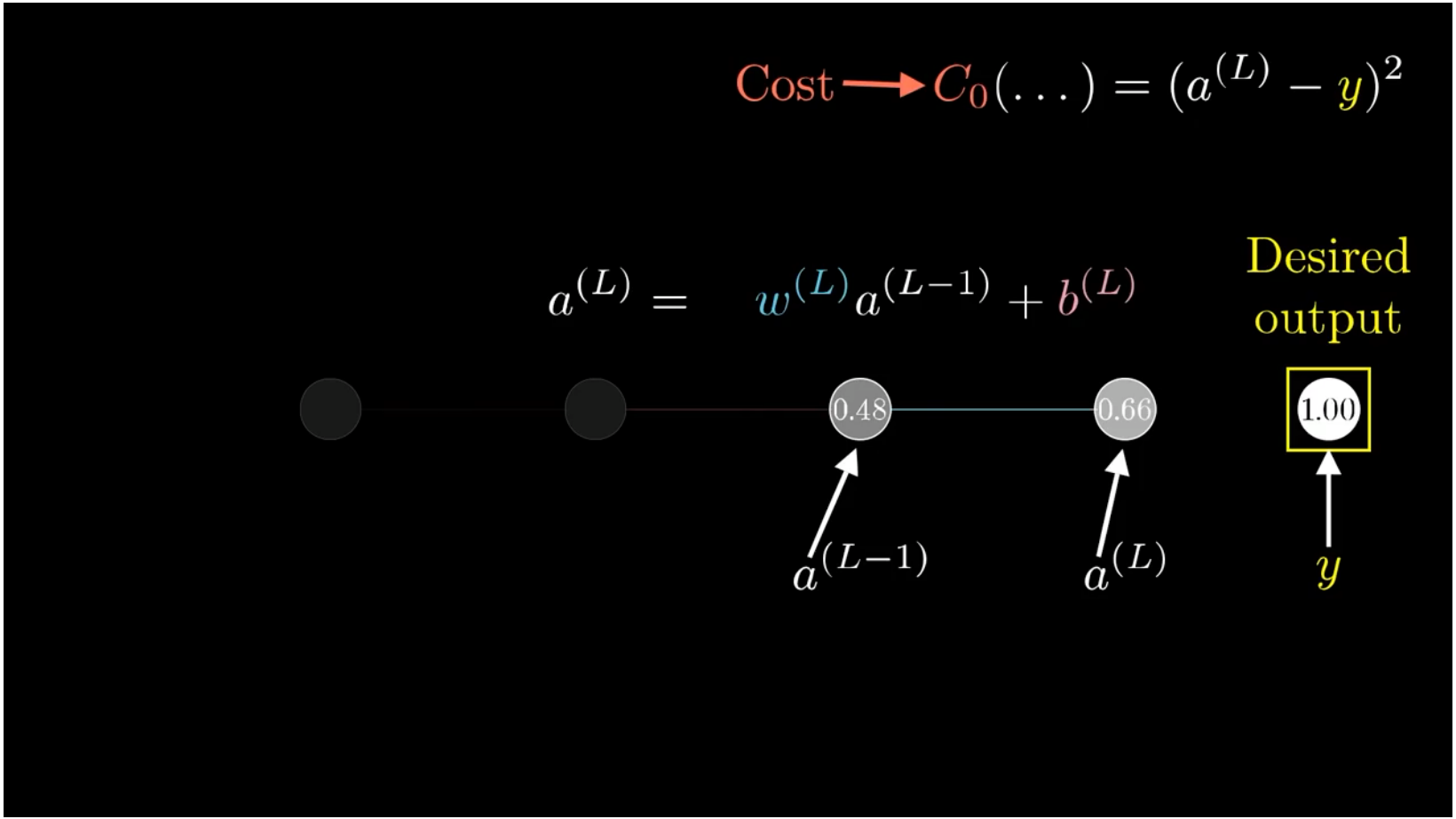

784
0
1
2
3
4
5
6
7
8
9

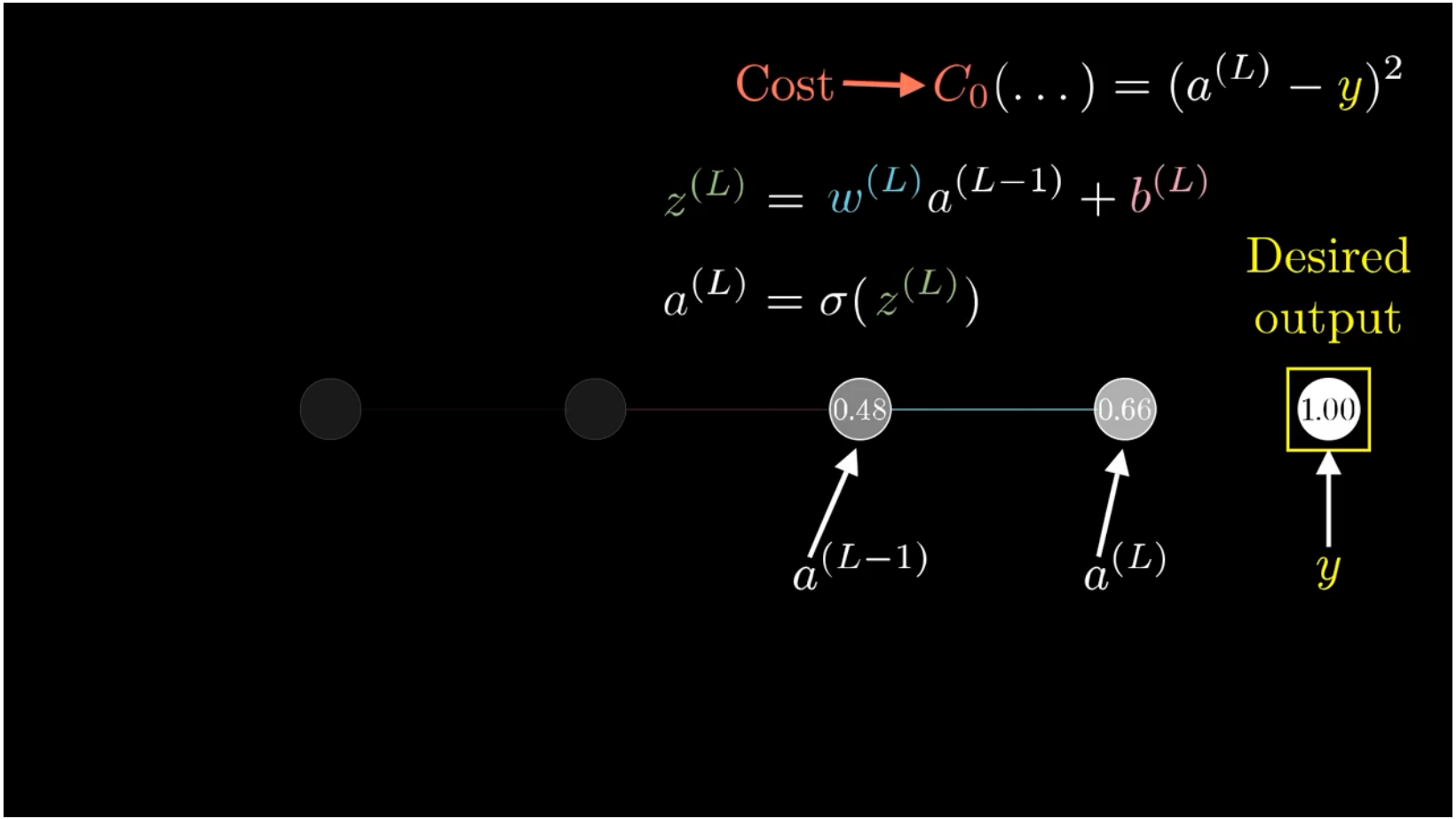
Seems very overwhelming…




0.48
0.66
$a^{(L-1)}$
$a^{(L)}$


$C_0(\dots) = (a^{(L)} - y)^2$
Desired output
0.48
0.66
1.00
$a^{(L-1)}$
$a^{(L)}$
y

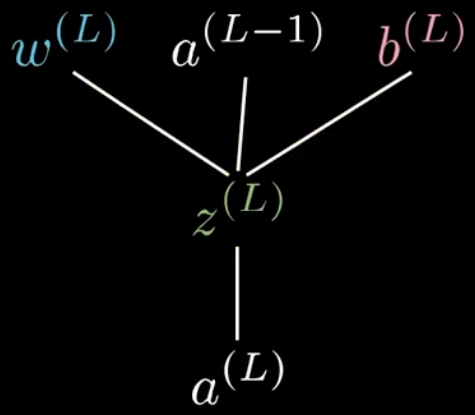

Cost → $C_0(\dots) = (a^{(L)} - y)^2$
$a^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$
Desired output
0.48
0.66
1.00
$a^{(L-1)}$
$a^{(L)}$
y

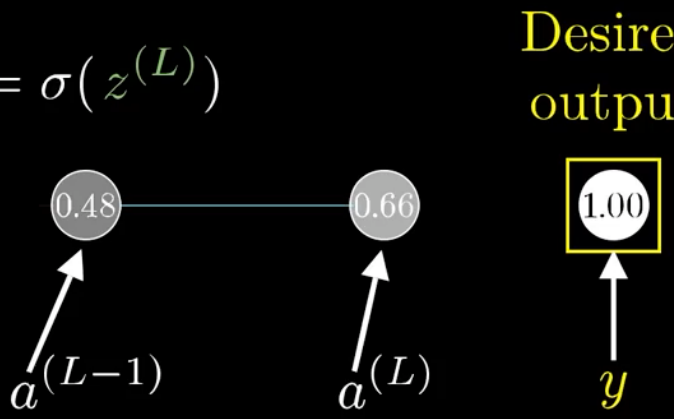

Cost → $C_0(\dots) = (a^{(L)} - y)^2$
$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$
$a^{(L)} = \sigma(z^{(L)})$
Desired output
0.48
0.66
1.00
$a^{(L-1)}$
$a^{(L)}$
y

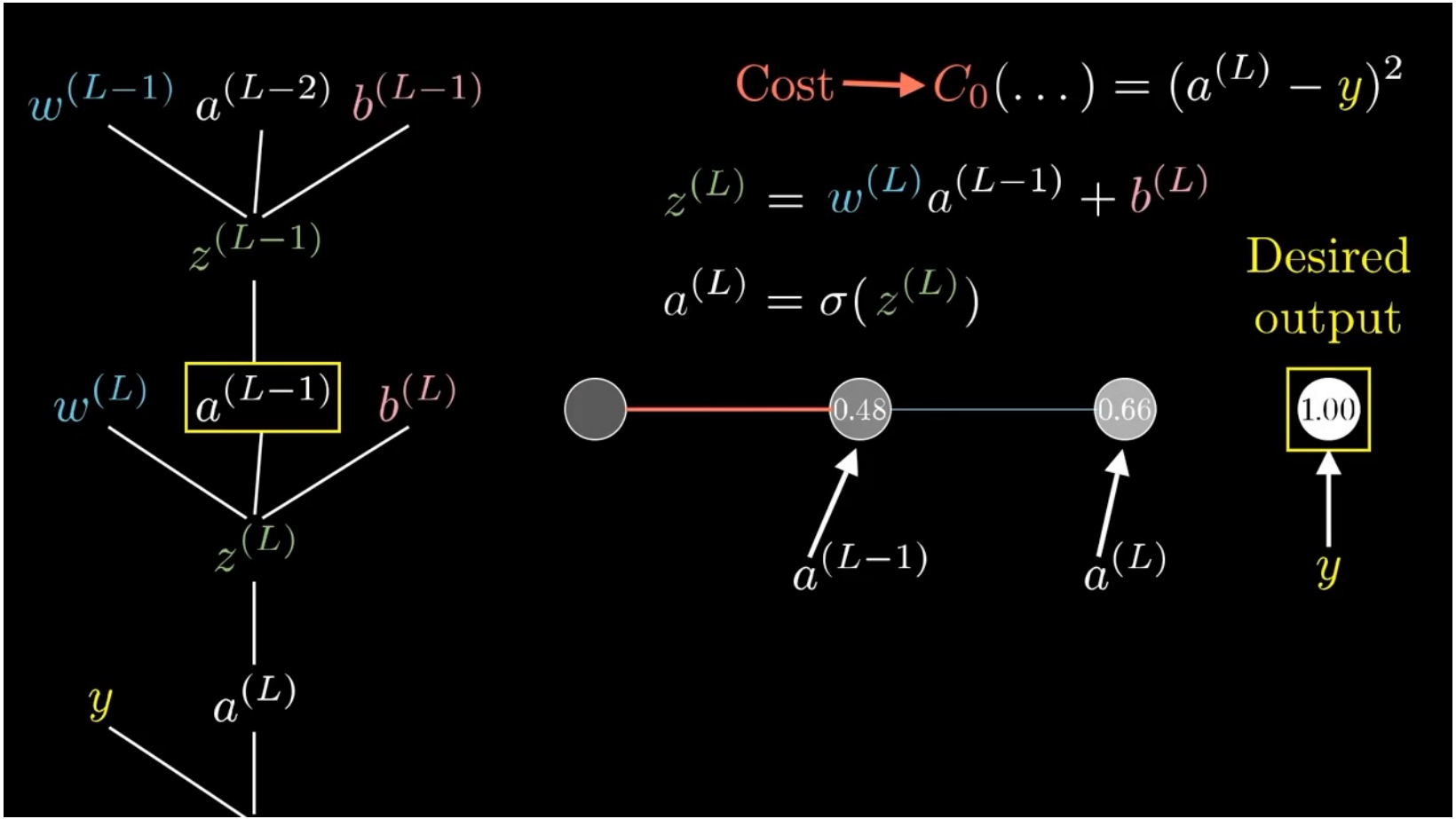
$$\text{Cost} \longrightarrow C_0(\dots) = (a^{(L)} - y)^2$$

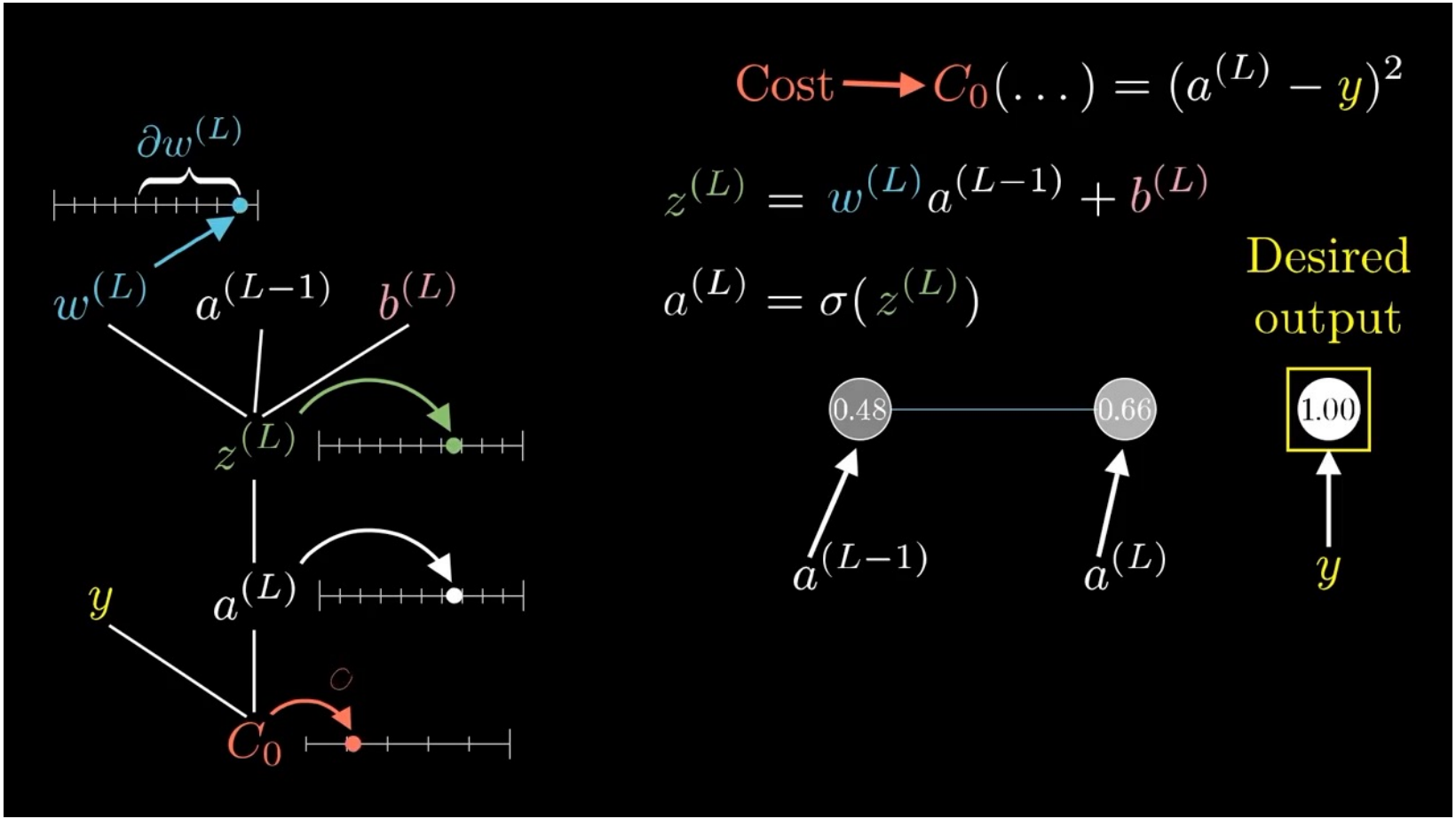
$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

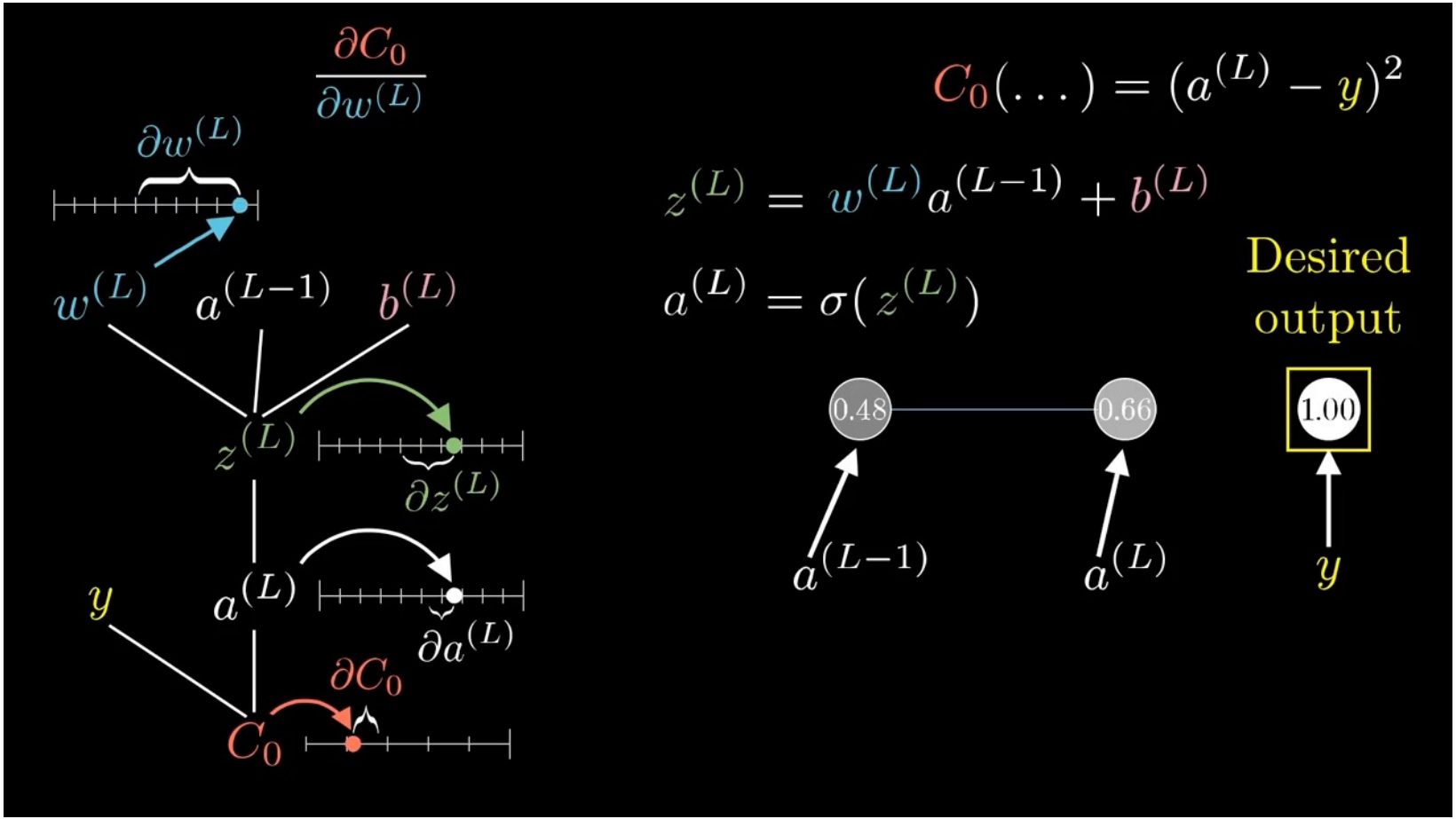
$$a^{(L)} = \sigma(z^{(L)})$$






$w^{(L-1)}$ $a^{(L-2)}$ $b^{(L-1)}$
$z^{(L-1)}$
$w^{(L)}$ $a^{(L-1)}$ $b^{(L)}$
$z^{(L)}$
y $a^{(L)}$
Cost $C_0(\dots) = (a^{(L)} - y)^2$
$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$
$a^{(L)} = \sigma(z^{(L)})$
Desired output
0.48
0.66
1.00
$a^{(L-1)}$
$a^{(L)}$
y


$\text{Cost} \longrightarrow C_0(\dots) = (a^{(L)} - y)^2$
$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$
$a^{(L)} = \sigma(z^{(L)})$
Desired output
$\partial w^{(L)}$
$w^{(L)}$
$a^{(L-1)}$
$b^{(L)}$
$z^{(L)}$
$a^{(L)}$
y
C
C_0
0.48
0.66
1.00
$a^{(L-1)}$
$a^{(L)}$
y

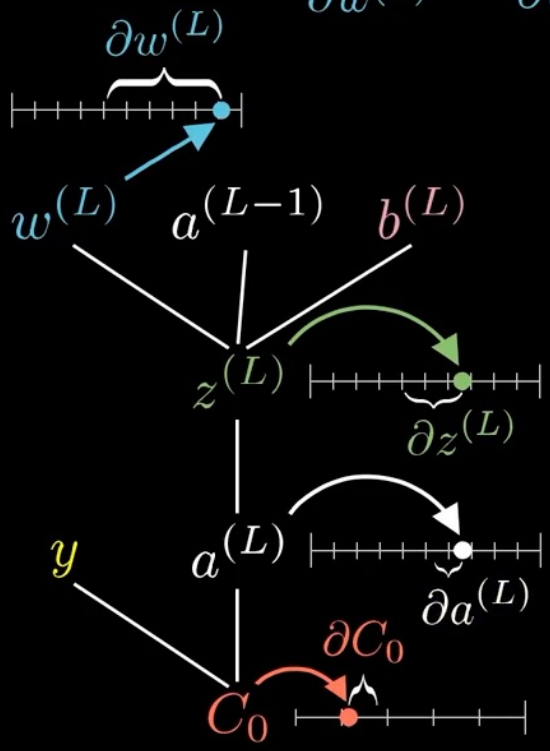

$\frac{\partial C_0}{\partial w^{(L)}}$
$C_0(\dots) = (a^{(L)} - y)^2$
$\partial w^{(L)}$
$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$
$w^{(L)}$
$a^{(L-1)}$
$b^{(L)}$
$a^{(L)} = \sigma(z^{(L)})$
Desired output
$z^{(L)}$
$\partial z^{(L)}$
0.48
0.66
1.00
$a^{(L-1)}$
$a^{(L)}$
y
$a^{(L)}$
$\partial a^{(L)}$
y
∂C_0
C_0

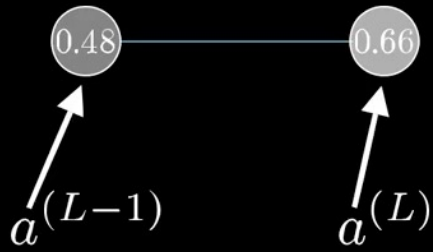
$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}}$$

$$C_0(\dots) = (a^{(L)} - y)^2$$

$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

$$a^{(L)} = \sigma(z^{(L)})$$





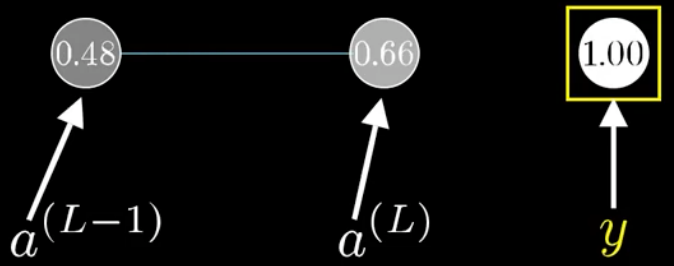
Desired output

$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}}$$

$$C_0 = (a^{(L)} - y)^2$$

$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

$$a^{(L)} = \sigma(z^{(L)})$$



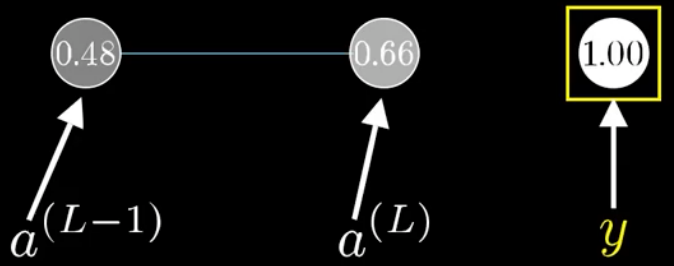
$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}}$$

$$C_0 = (a^{(L)} - y)^2$$

$$\frac{\partial C0}{\partial a^{(L)}} = 2(a^{(L)} - y)$$

$$a^{(L)} = \sigma(z^{(L)})$$

$$\frac{\partial a^{(L)}}{\partial z^{(L)}} = \sigma'(z^{(L)})$$



$$\frac{\partial z^{(L)}}{\partial w^{(L)}} = a^{(L-1)}$$

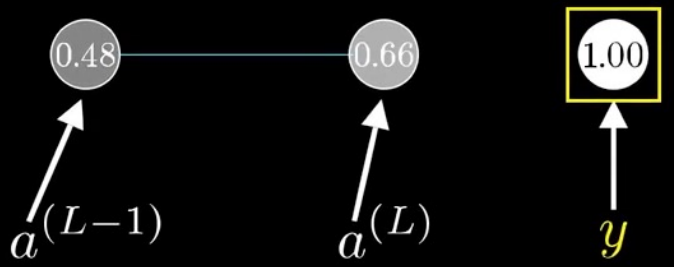
$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}} = a^{(L-1)} \sigma'(z^{(L)}) 2(a^{(L)} - y)$$

$$C_0 = (a^{(L)} - y)^2$$

$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

$$a^{(L)} = \sigma(z^{(L)})$$



$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}} = a^{(L-1)} \sigma'(z^{(L)}) 2(a^{(L)} - y)$$

$$C_0 = (a^{(L)} - y)^2$$

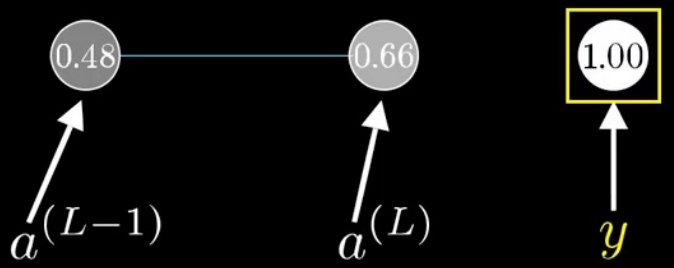
Average of all training examples

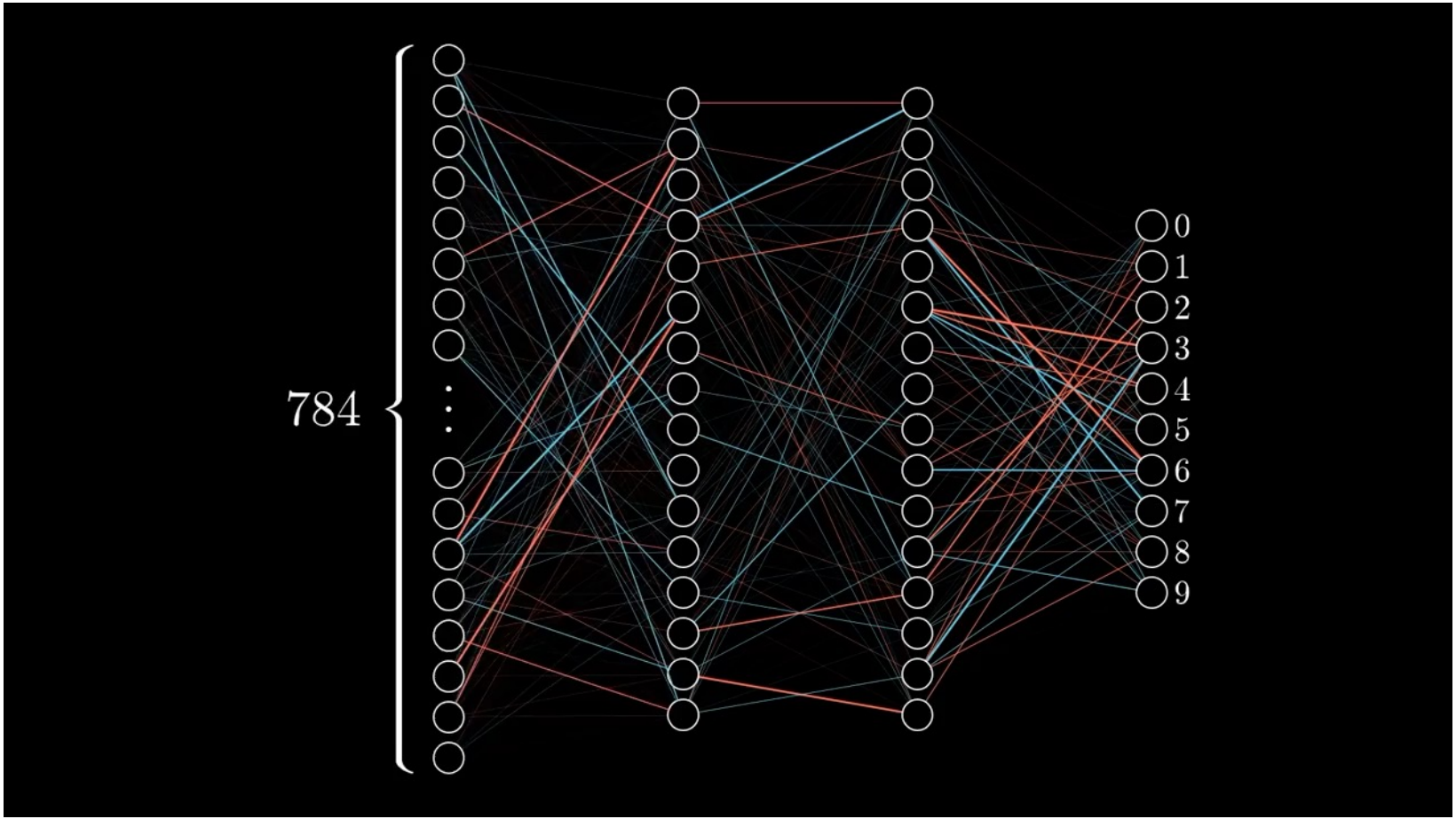
$$\underbrace{\frac{\partial C}{\partial w^{(L)}}} = \overbrace{\frac{1}{n} \sum_{k=0}^{n-1} \frac{\partial C_k}{\partial w^{(L)}}}$$

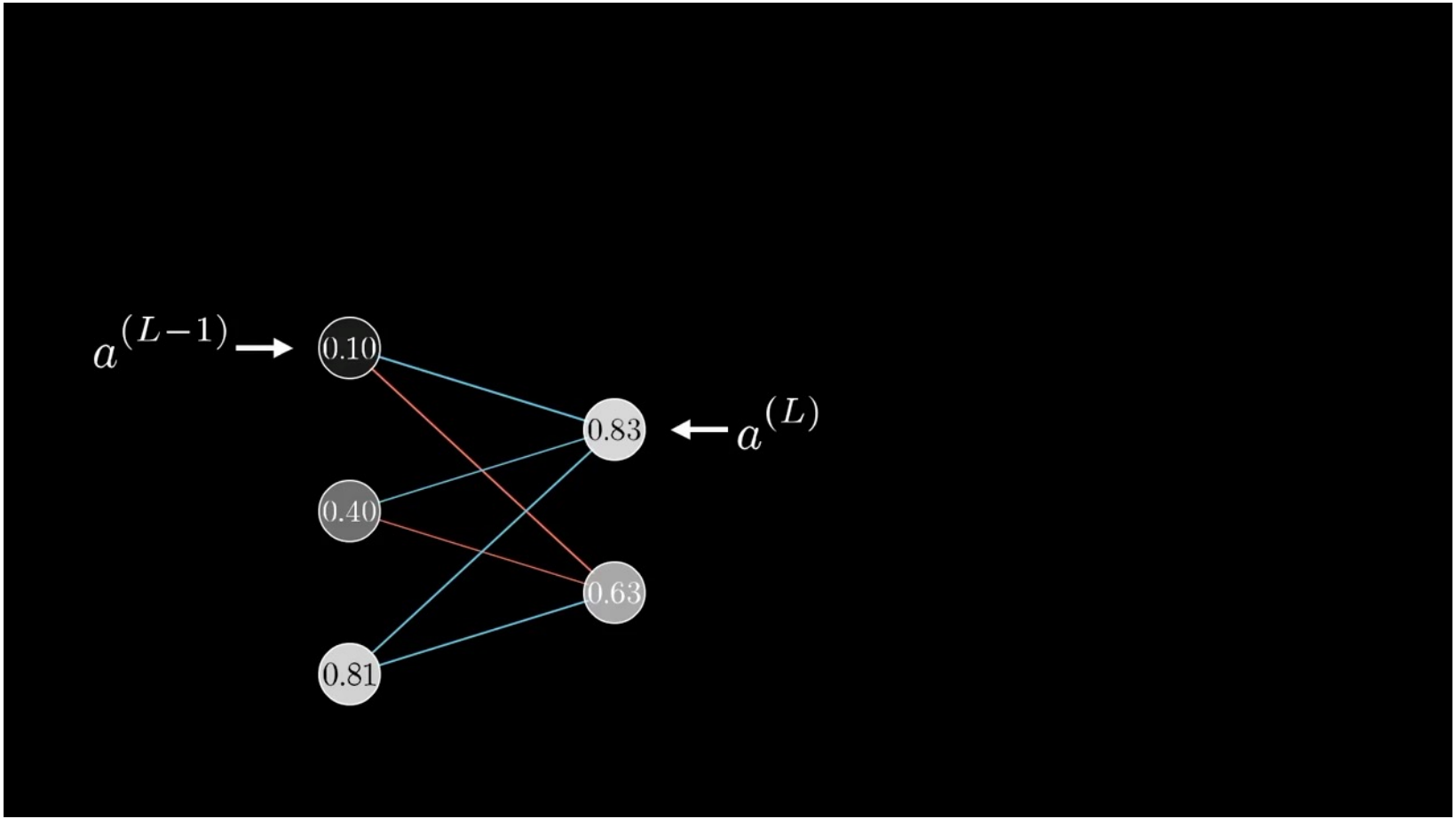
Derivative of full cost function

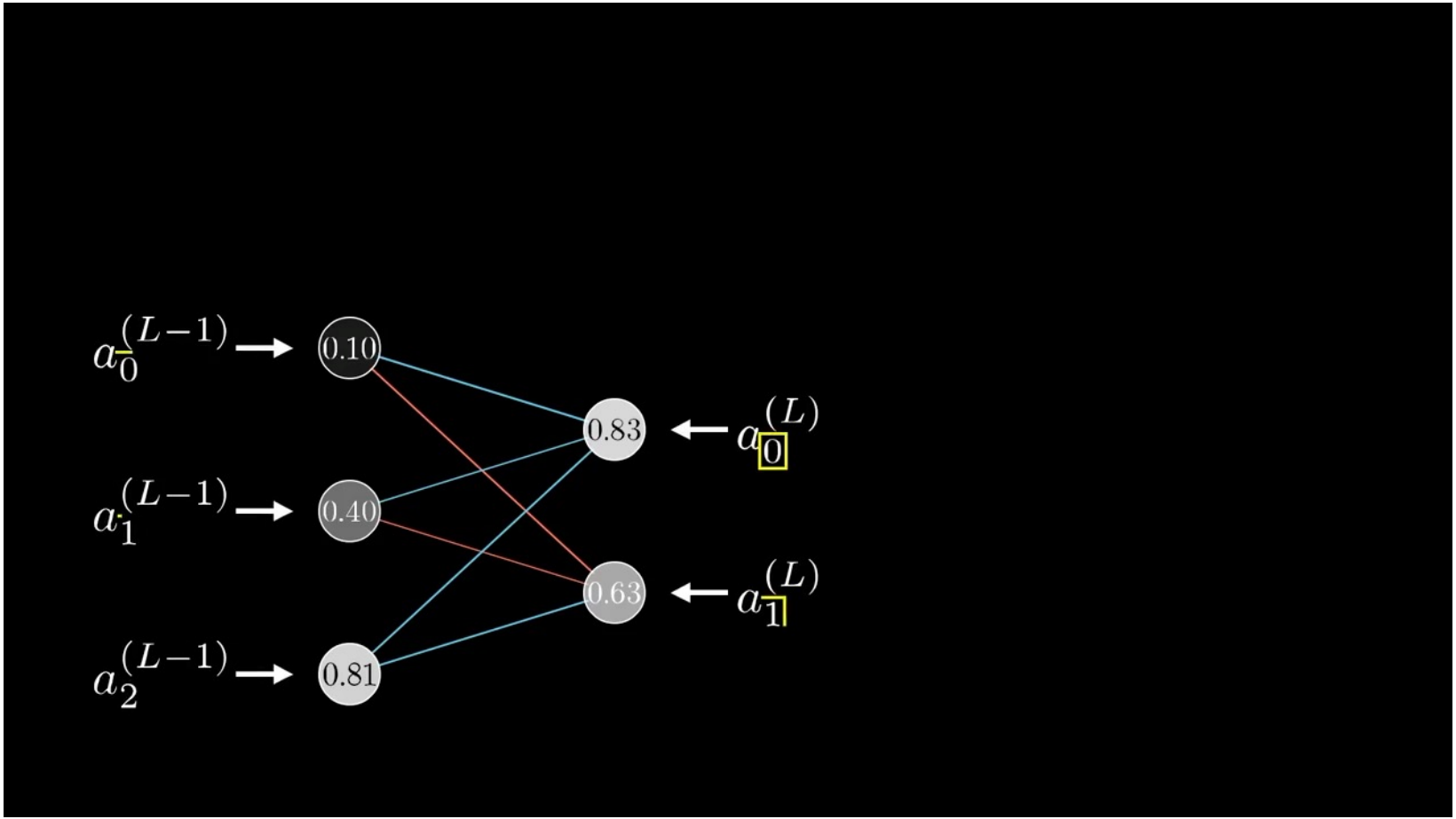
$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

$$a^{(L)} = \sigma(z^{(L)})$$

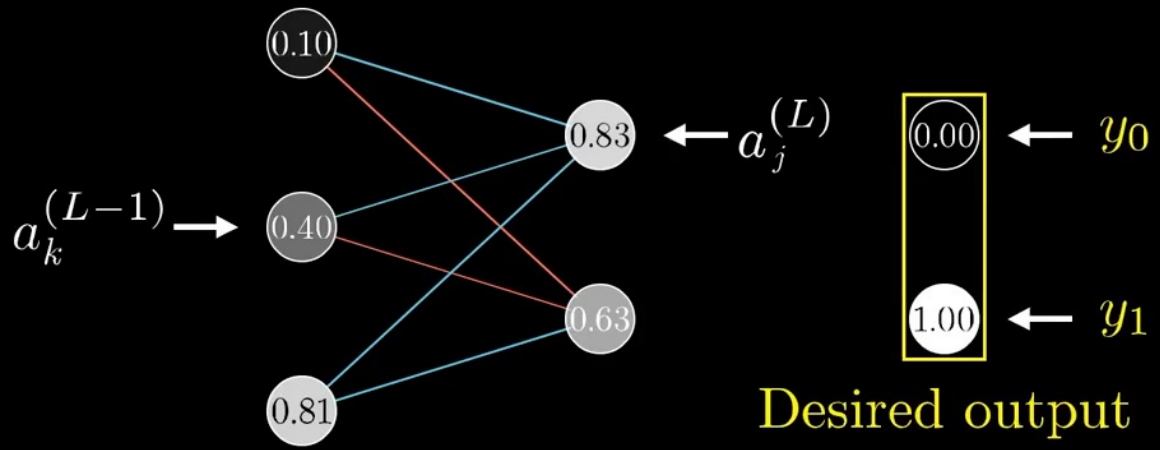



784
0
1
2
3
4
5
6
7
8
9

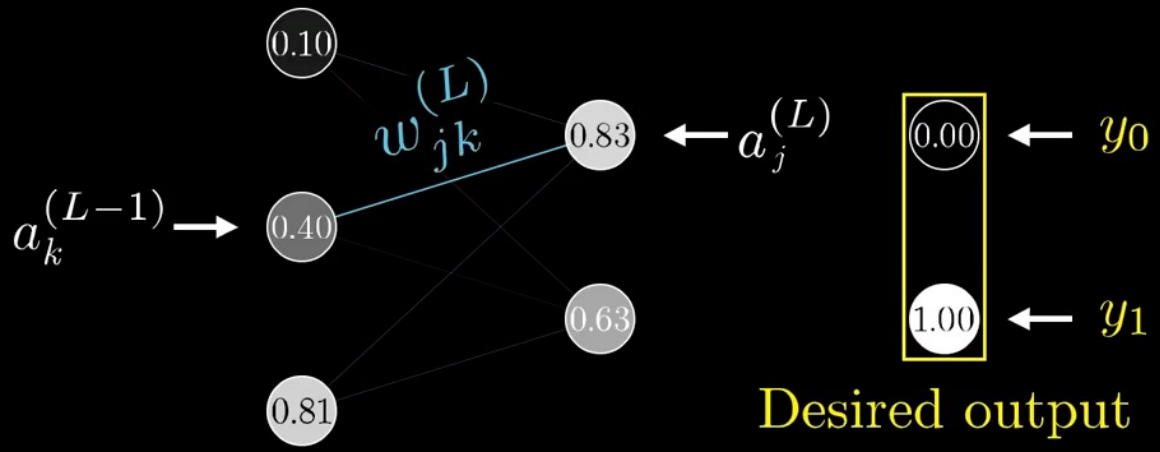

$a^{(L-1)}$
0.10
0.40
0.81
0.83
0.63
$a^{(L)}$


$a_0^{(L-1)}$
$a_1^{(L-1)}$
$a_2^{(L-1)}$
0.10
0.40
0.81
0.83
0.63
$a_0^{(L)}$
$a_1^{(L)}$

$$C_0 = \sum_{j=0}^{n_L-1} (a_j^{(L)} - y_j)^2$$

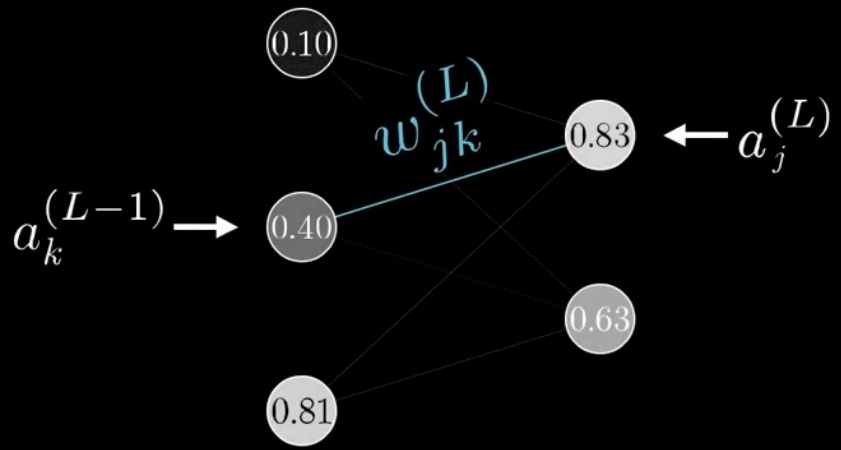


$$C_0 = \sum_{j=0}^{n_L - 1} (a_j^{(L)} - y_j)^2$$



$$z_j^{(L)} = w_{j0}^{(L)} a_0^{(L-1)} + w_{j1}^{(L)} a_1^{(L-1)} + w_{j2}^{(L)} a_2^{(L-1)} + b_j^{(L)}$$

$$a_j^{(L)} = \sigma(z_j^{(L)})$$

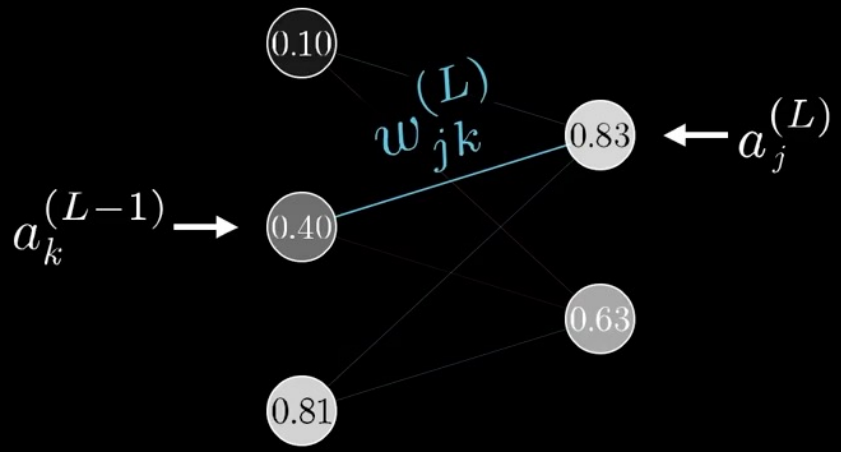


$$C_0 = \sum_{j=0}^{n_L - 1} (a_j^{(L)} - y_j)^2$$

$$\frac{\partial C_0}{\partial w_{jk}^{(L)}} = \frac{\partial z_j^{(L)}}{\partial w_{jk}^{(L)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}}$$

$$z_j^{(L)} = \cdots + w_{jk}^{(L)} a_k^{(L-1)} + \cdots$$

$$a_j^{(L)} = \sigma(z_j^{(L)})$$

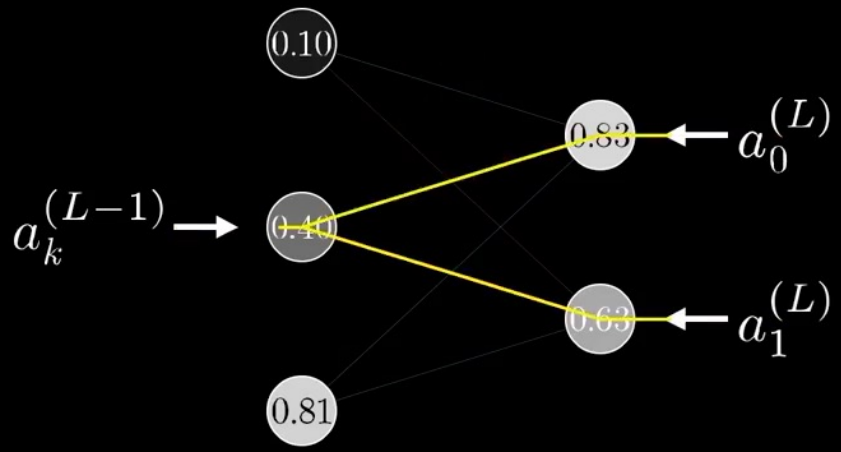


$$C_0 = \sum_{j=0}^{n_L - 1} (a_j^{(L)} - y_j)^2$$

$$\frac{\partial C_0}{\partial a_k^{(L-1)}} = \sum_{j=0}^{n_L-1} \frac{\partial z_j^{(L)}}{\partial a_k^{(L-1)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}}$$

$$z_j^{(L)} = \cdots + w_{jk}^{(L)} a_k^{(L-1)} + \cdots$$

$$a_j^{(L)} = \sigma(z_j^{(L)})$$



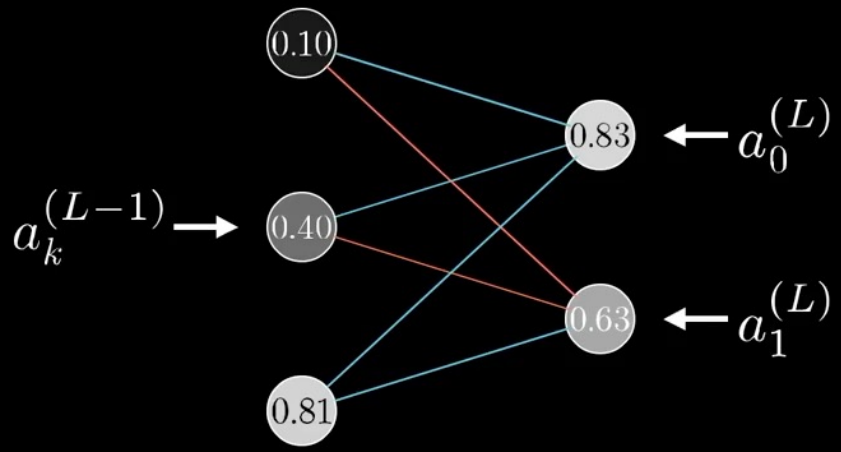
$$C_0 = \sum_{j=0}^{n_L-1} (a_j^{(L)} - y_j)^2$$

$$\frac{\partial C_0}{\partial a_k^{(L-1)}} = \underbrace{\sum_{j=0}^{n_L-1} \frac{\partial z_j^{(L)}}{\partial a_k^{(L-1)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}}}_{\text{Sum over layer L}}$$

$$z_j^{(L)} = \cdots + w_{jk}^{(L)} a_k^{(L-1)} + \cdots$$

$$a_j^{(L)} = \sigma(z_j^{(L)})$$

$$C_0 = \sum_{j=0}^{n_L-1} (a_j^{(L)} - y_j)^2$$

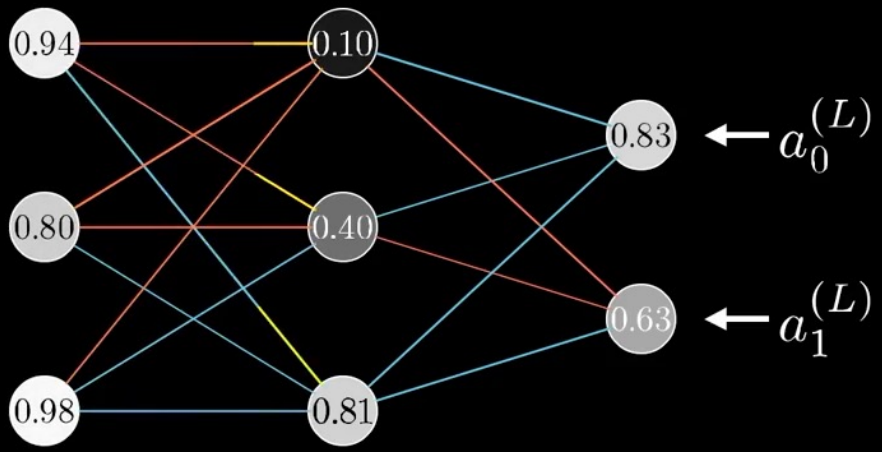


$$\frac{\partial C_0}{\partial a_k^{(L-1)}} = \underbrace{\sum_{j=0}^{n_L-1} \frac{\partial z_j^{(L)}}{\partial a_k^{(L-1)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial C_0}{\partial a_j^{(L)}}}_{\text{Sum over layer L}}$$

$$z_j^{(L)} = \cdots + w_{jk}^{(L)} a_k^{(L-1)} + \cdots$$

$$a_j^{(L)} = \sigma(z_j^{(L)})$$

$$C_0 = \sum_{j=0}^{n_L-1} (a_j^{(L)} - y_j)^2$$



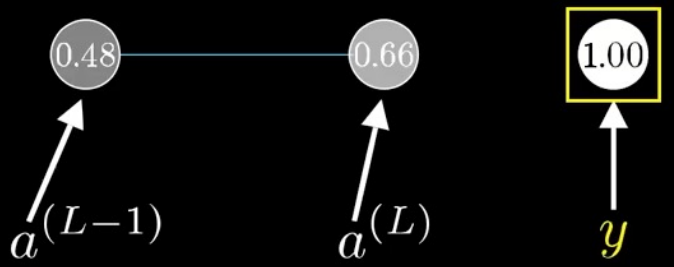
$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C0}{\partial a^{(L)}} = a^{(L-1)} \sigma'(z^{(L)}) 2(a^{(L)} - y)$$

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial w^{(1)}} \\ \frac{\partial C}{\partial b^{(1)}} \\ \vdots \\ \frac{\partial C}{\partial w^{(L)}} \\ \frac{\partial C}{\partial b^{(L)}} \end{bmatrix}$$

$$C_0 = (a^{(L)} - y)^2$$

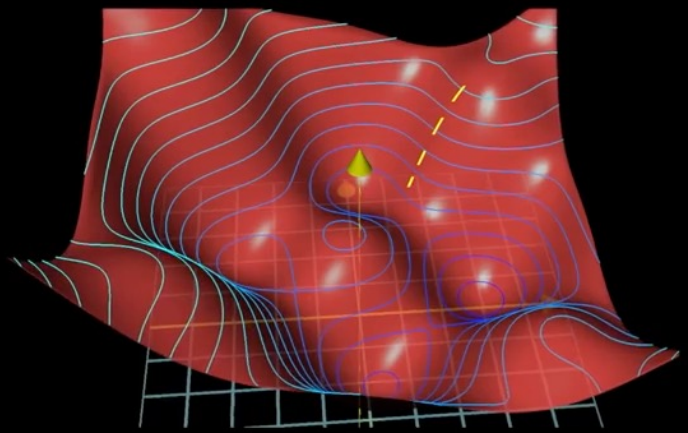
$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

$$a^{(L)} = \sigma(z^{(L)})$$



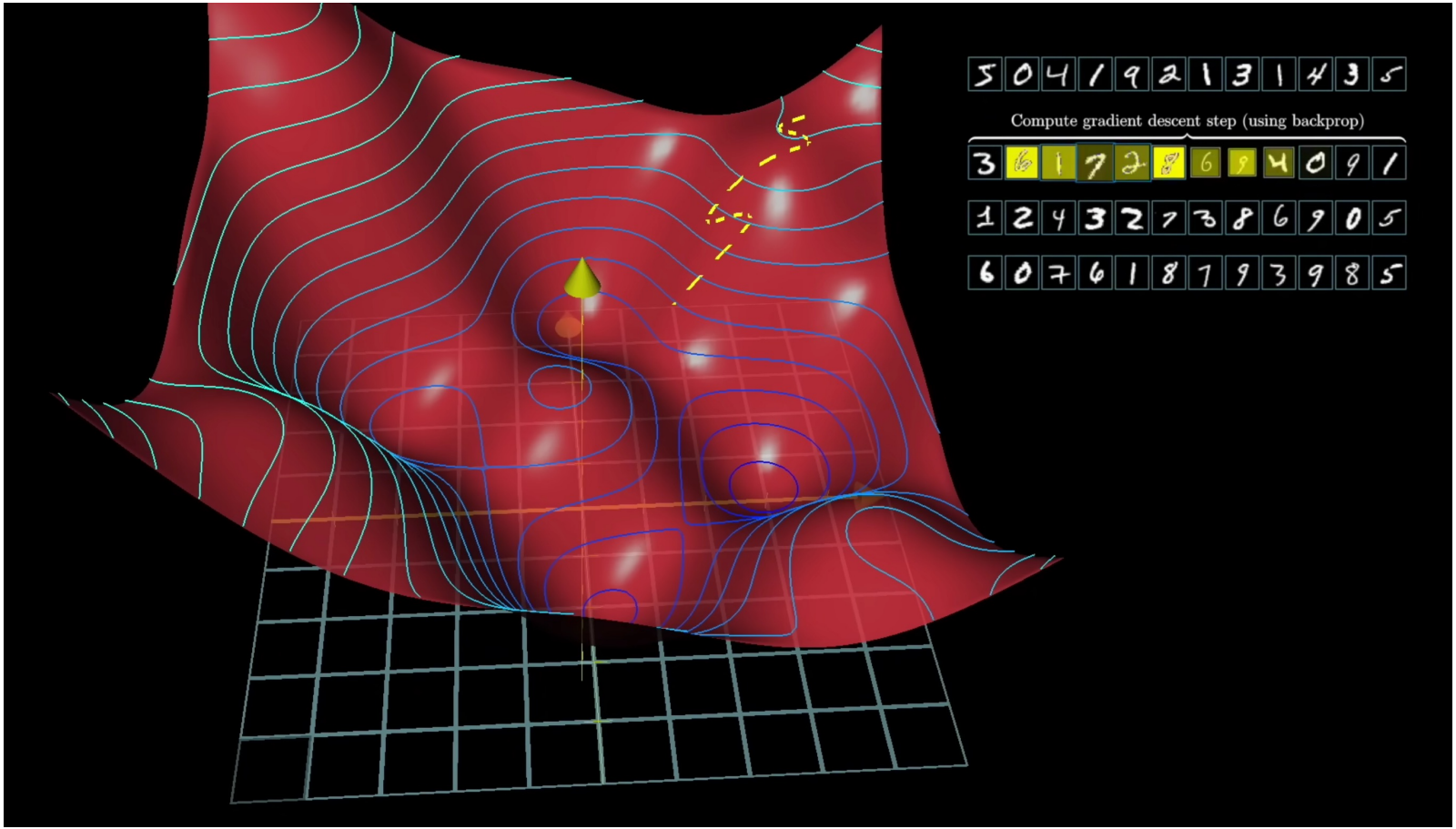
$$\frac{\partial C}{\partial w_{jk}^{(l)}} = a_k^{(l-1)} \sigma'(z_j^{(l)}) \boxed{\frac{\partial C}{\partial a_j^{(l)}}}$$

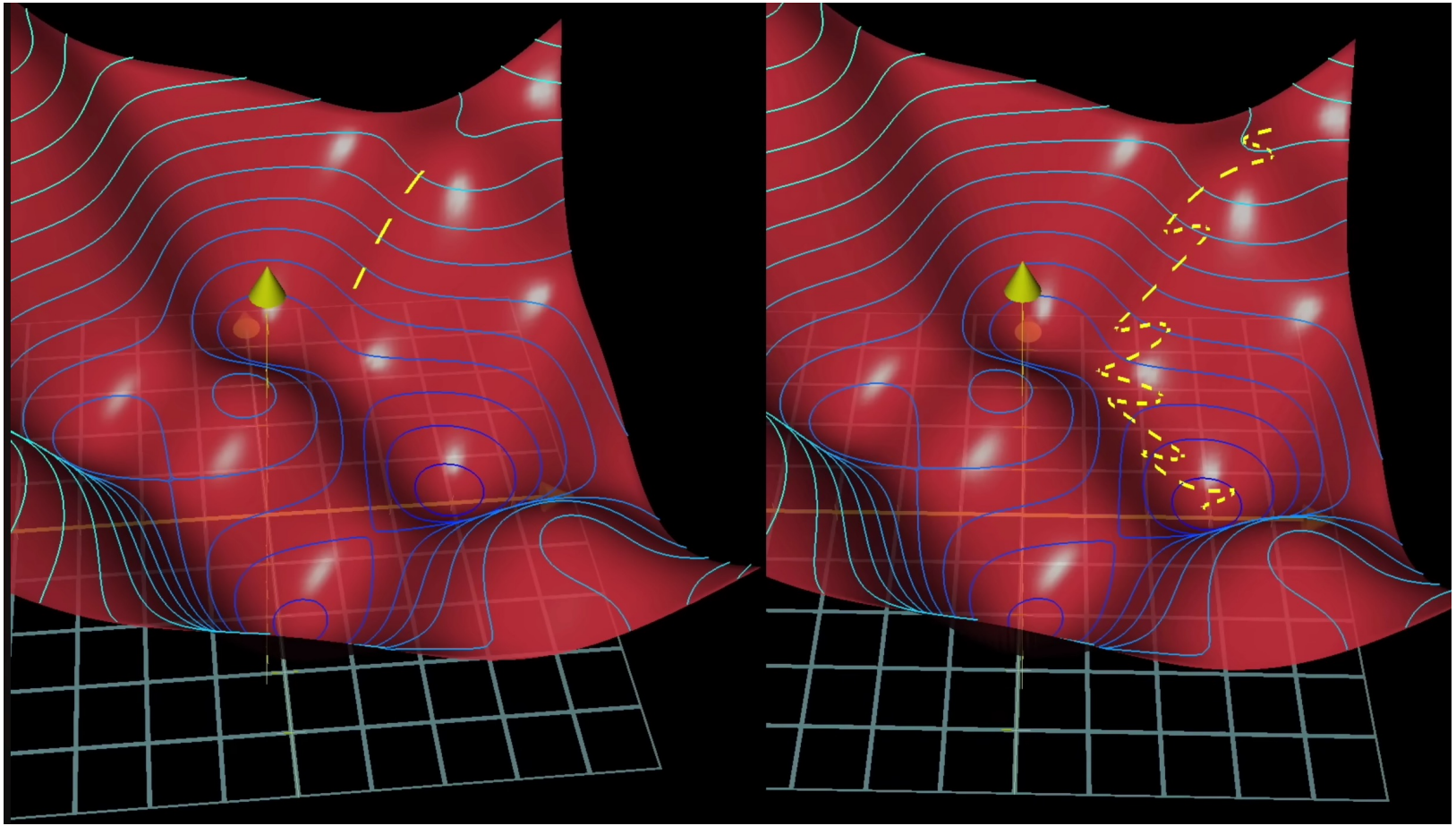
$$\nabla C \leftarrow \begin{cases} \frac{\partial C}{\partial w_{jk}^{(l)}} = a_k^{(l-1)} \sigma'(z_j^{(l)}) \frac{\partial C}{\partial a_j^{(l)}} \\ \boxed{\begin{array}{c} \sum_{j=0}^{n_{l+1}-1} w_{jk}^{(l+1)} \sigma'(z_j^{(l+1)}) \frac{\partial C}{\partial a_j^{(l+1)}} \\ \text{or} \\ 2(a_j^{(L)} - y_j) \end{array}} \end{cases}$$



A little recap about ***Backprop…***


"Mini-batches"


Compute gradient descent step (using backprop)



Summary in words

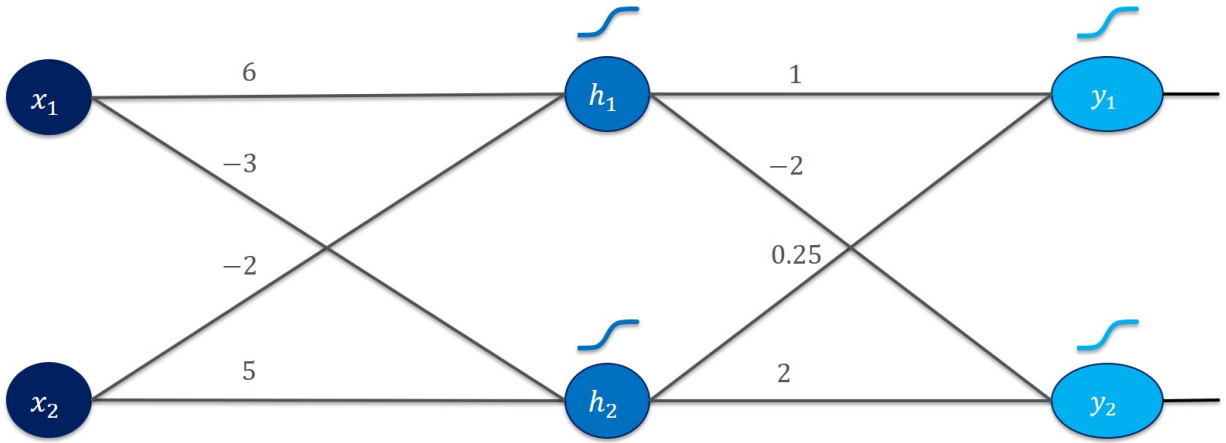
- Backprop determine how a single training example nudge the weights and biases

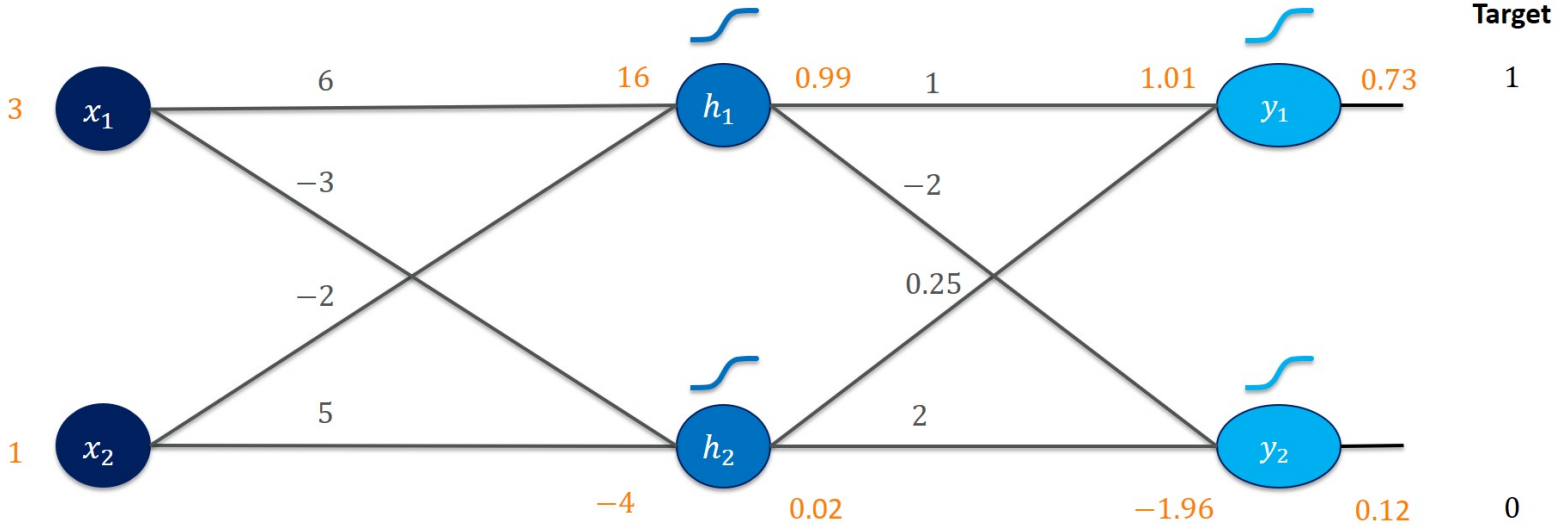
- Gradient descent step involve doing this for x1000000 of times
    - But that computationally slow…

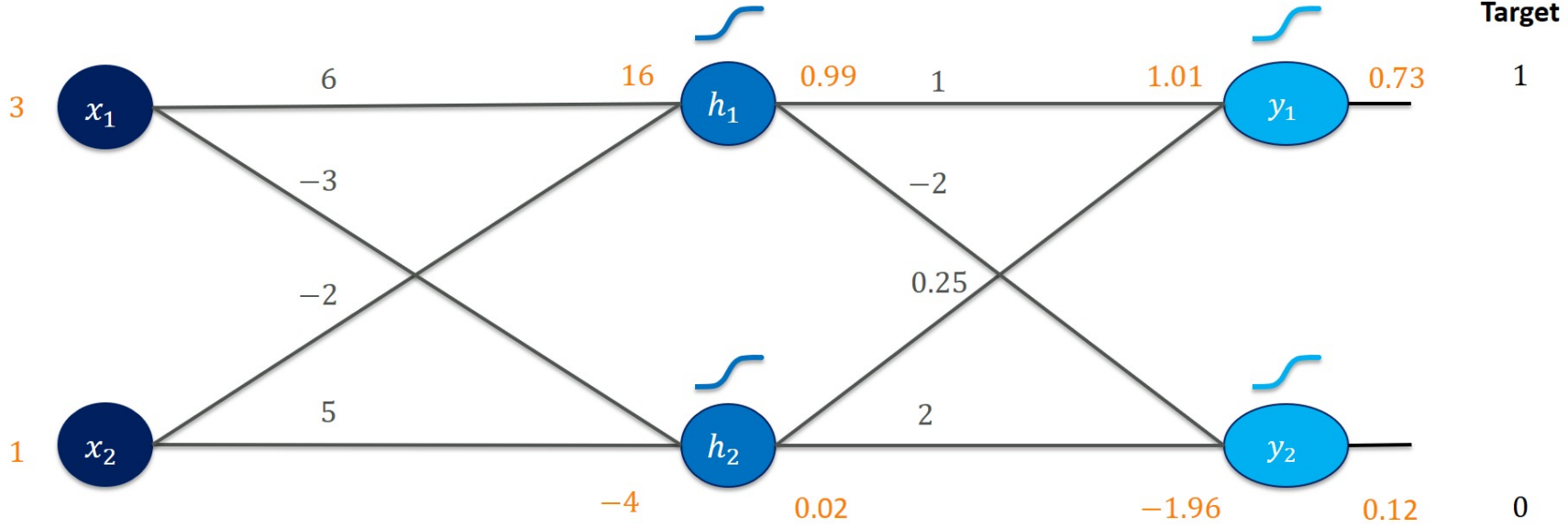
- Introduce mini-batches
    - Stochastic gradient descent

Examples work through

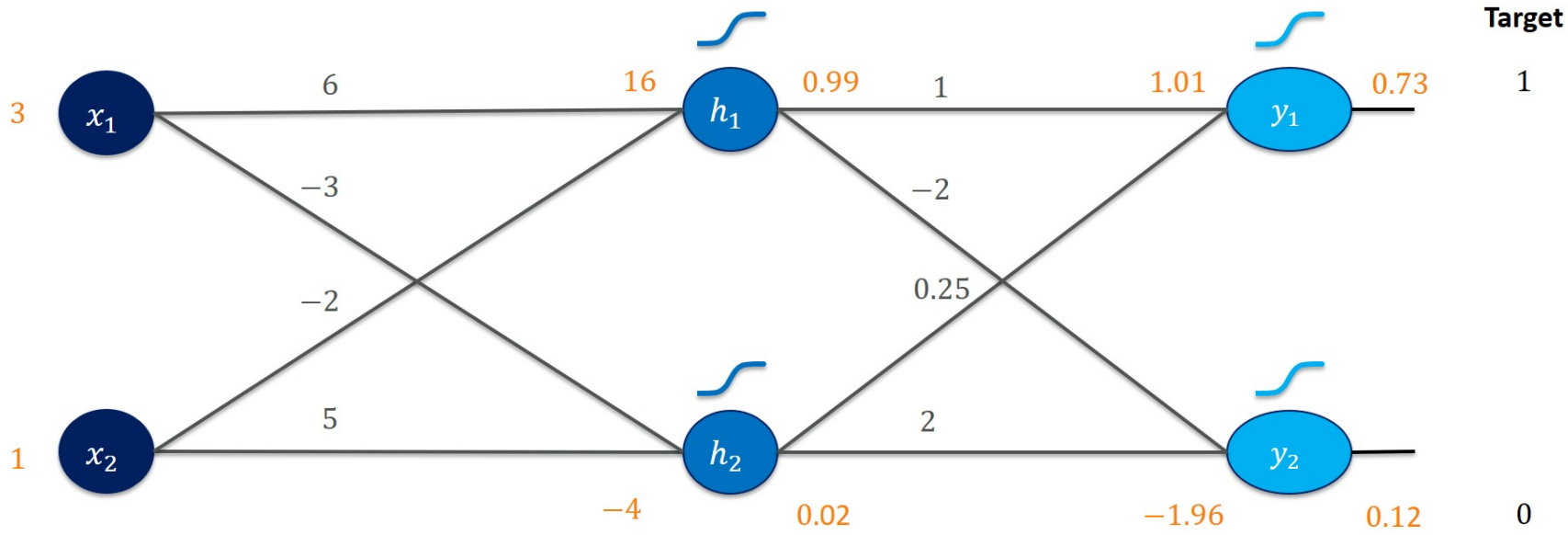
	Inputs		Targets	
Training data	x_1	x_2	t_1	t_2
	3	1	1	0
	-1	4	0	1

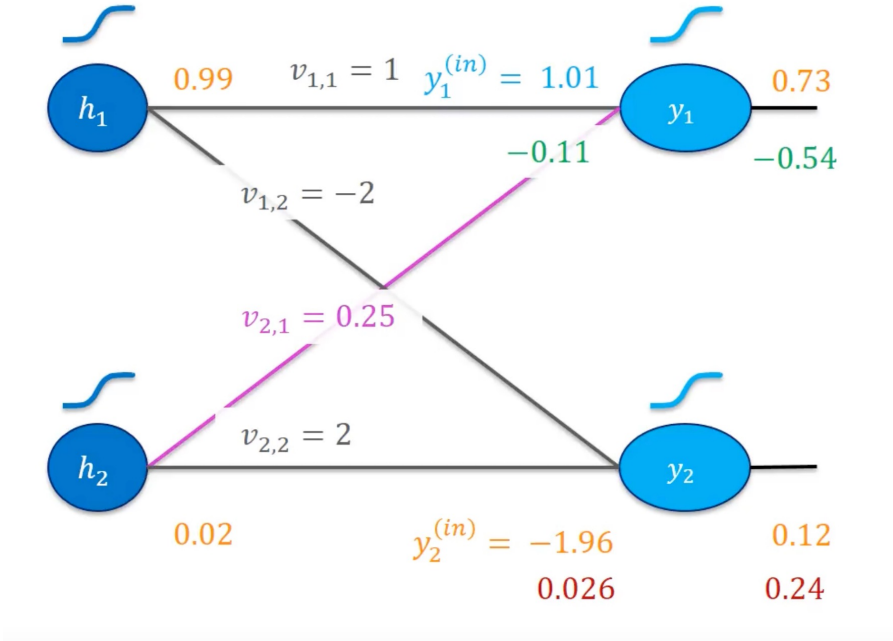


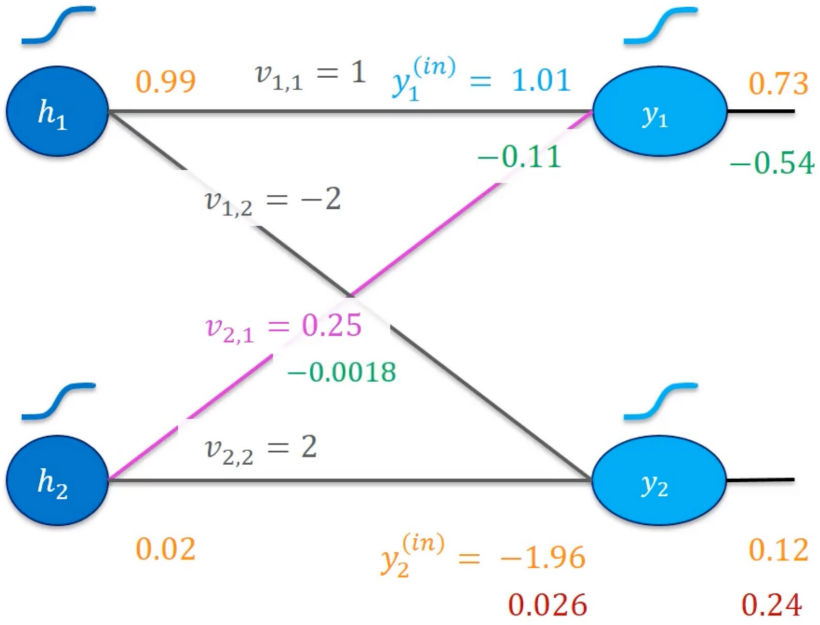

Target
3
x_1
6
16
h_1
0.99
1
1.01
y_1
0.73
1
−3
−2
−2
0.25
1
x_2
5
h_2
2
y_2
−4
0.02
−1.96
0.12
0


Target
3
x_1
1
x_2
6
−3
−2
5
16
−4
h_1
h_2
0.99
0.02
1
−2
0.25
2
1.01
−1.96
y_1
y_2
0.73
0.12
1
0

$$L(y_1, y_2) = (y_1 - t_1)^2 + (y_2 - t_2)^2 = e_1^2 + e_2^2$$




h_1
h_2
y_1
y_2
0.99
$v_{1,1} = 1$
$y_1^{(in)} = 1.01$
0.73
−0.54
−0.11
$v_{1,2} = -2$
$v_{2,1} = 0.25$
$v_{2,2} = 2$
0.02
$y_2^{(in)} = -1.96$
0.12
0.026
0.24

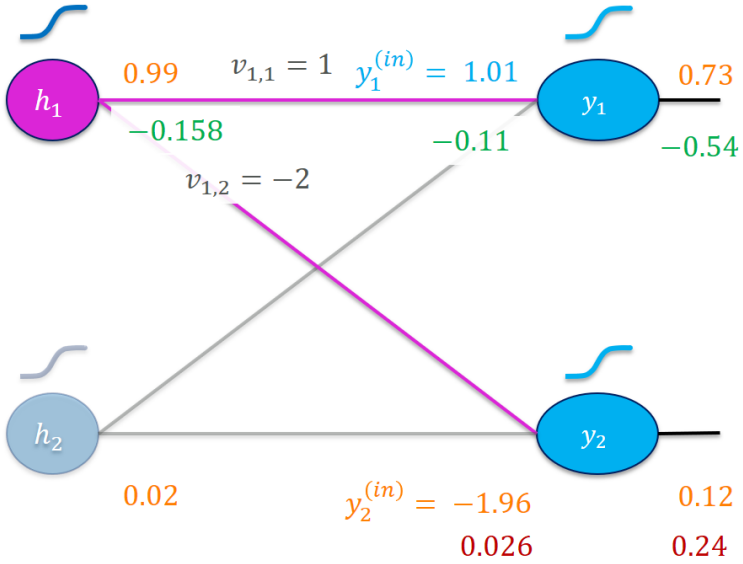


$$y_1^{(in)} = v_{1,1} \cdot h_1 + v_{2,1} \cdot h_2$$

$$\frac{\partial L}{\partial y_1^{(in)}} = -0.11$$

$$\frac{\partial L}{\partial v_{2,1}} = \frac{\partial L}{\partial y_1^{(in)}} \cdot \underbrace{\frac{\partial y_1^{(in)}}{\partial v_{2,1}}}_{h_2}$$

$$= -0.11 \cdot 0.02$$
$$= -0.0018$$



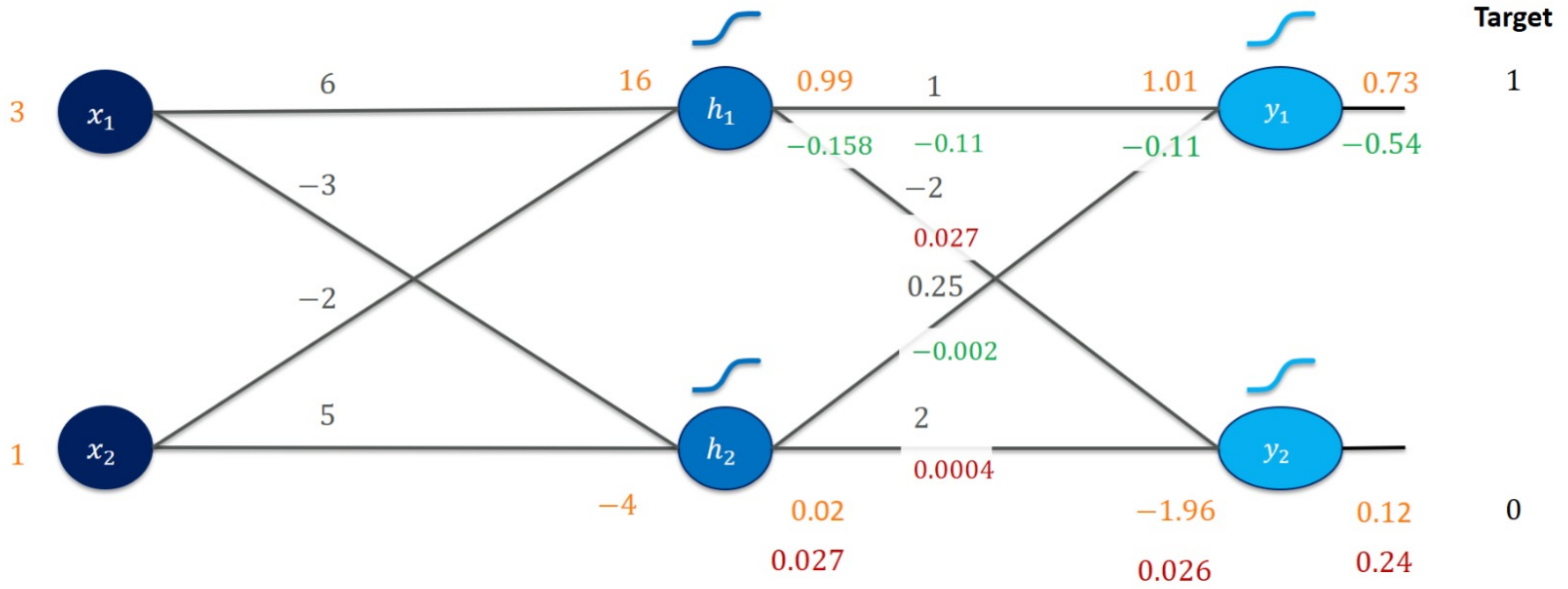
$$y_1^{(in)} = v_{1,1} \cdot h_1 + v_{2,1} \cdot h_2$$

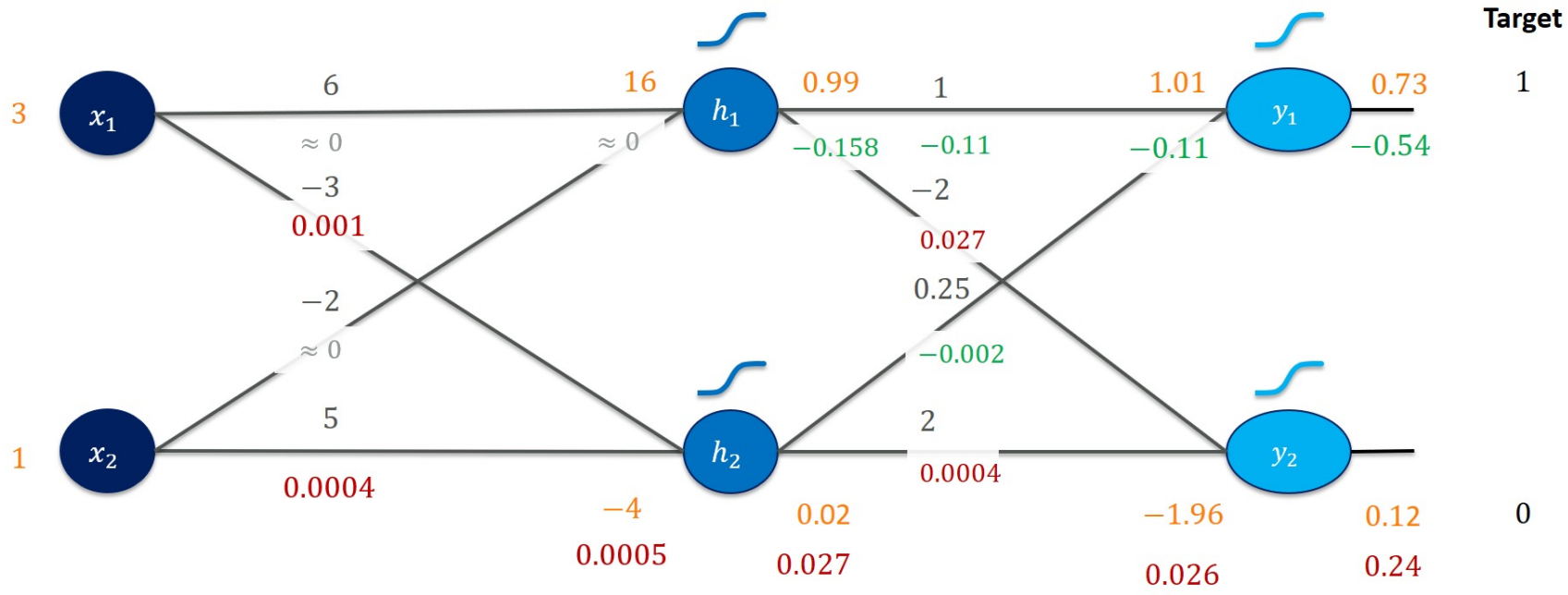
$$y_2^{(in)} = v_{1,2} \cdot h_1 + v_{2,2} \cdot h_2$$

$$\frac{\partial L}{\partial y_1^{(in)}} = -0.11 \qquad \frac{\partial L}{\partial y_2^{(in)}} = 0.026$$

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial y_1^{(in)}} \cdot \underbrace{\frac{\partial y_1^{(in)}}{\partial h_1}}_{v_{1,1}} + \frac{\partial L}{\partial y_2^{(in)}} \cdot \underbrace{\frac{\partial y_2^{(in)}}{\partial h_1}}_{v_{1,2}}$$

$$= -0.11 \cdot 1 + 0.026 \cdot (-2)$$
$$= -0.158$$


Target
3
x_1
6
16
h_1
0.99
1
1.01
y_1
0.73
1
−0.158
−0.11
−0.11
−0.54
−3
−2
0.027
0.25
−2
−0.002
1
x_2
5
h_2
2
y_2
0.0004
−4
0.02
0.027
−1.96
0.026
0.12
0.24
0



$$\nabla_w^{(1)} L = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0.0001 & 0.00004 & -0.11 & 0.027 & -0.002 & 0.0004 \\ \hline \end{array}$$

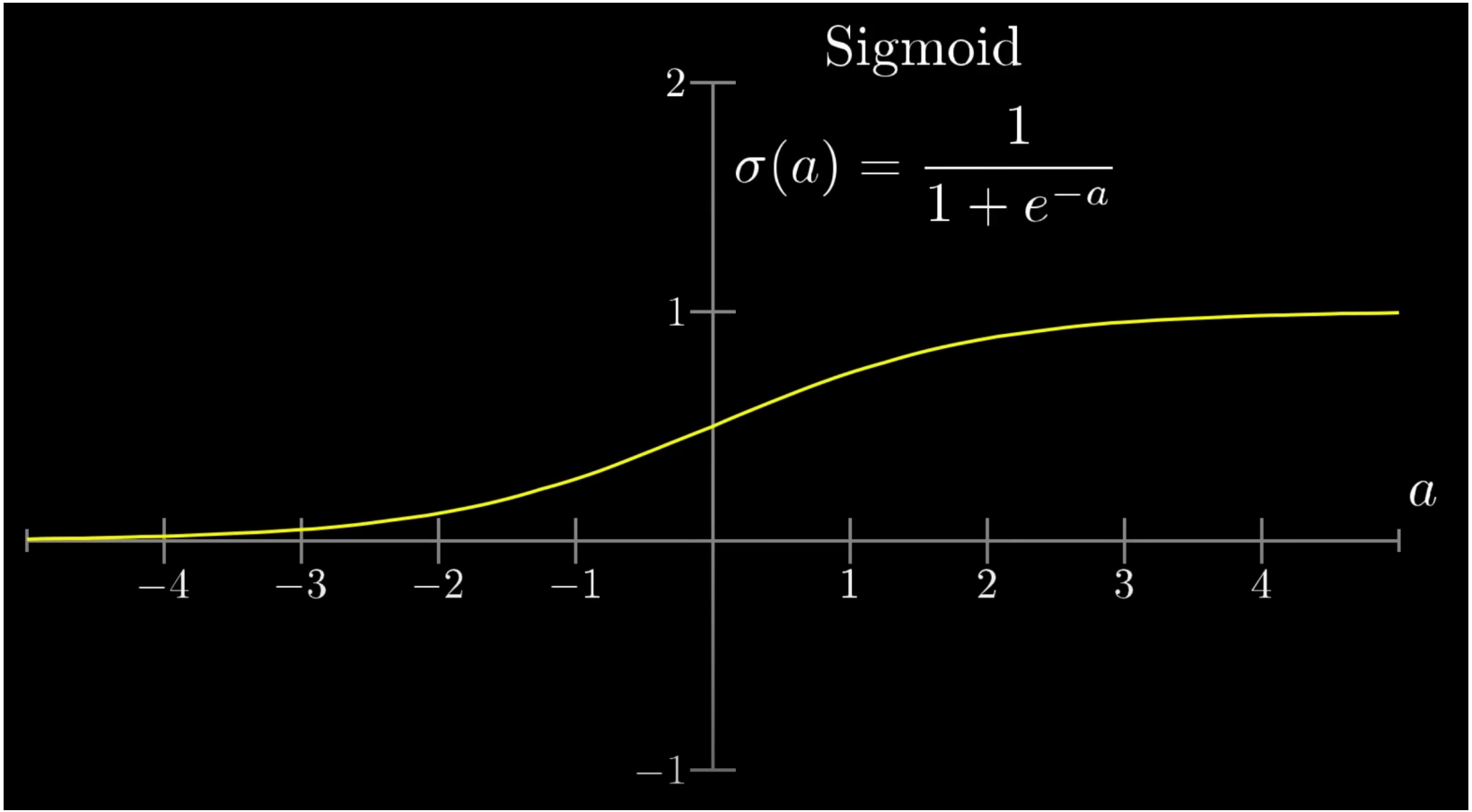
Activation function

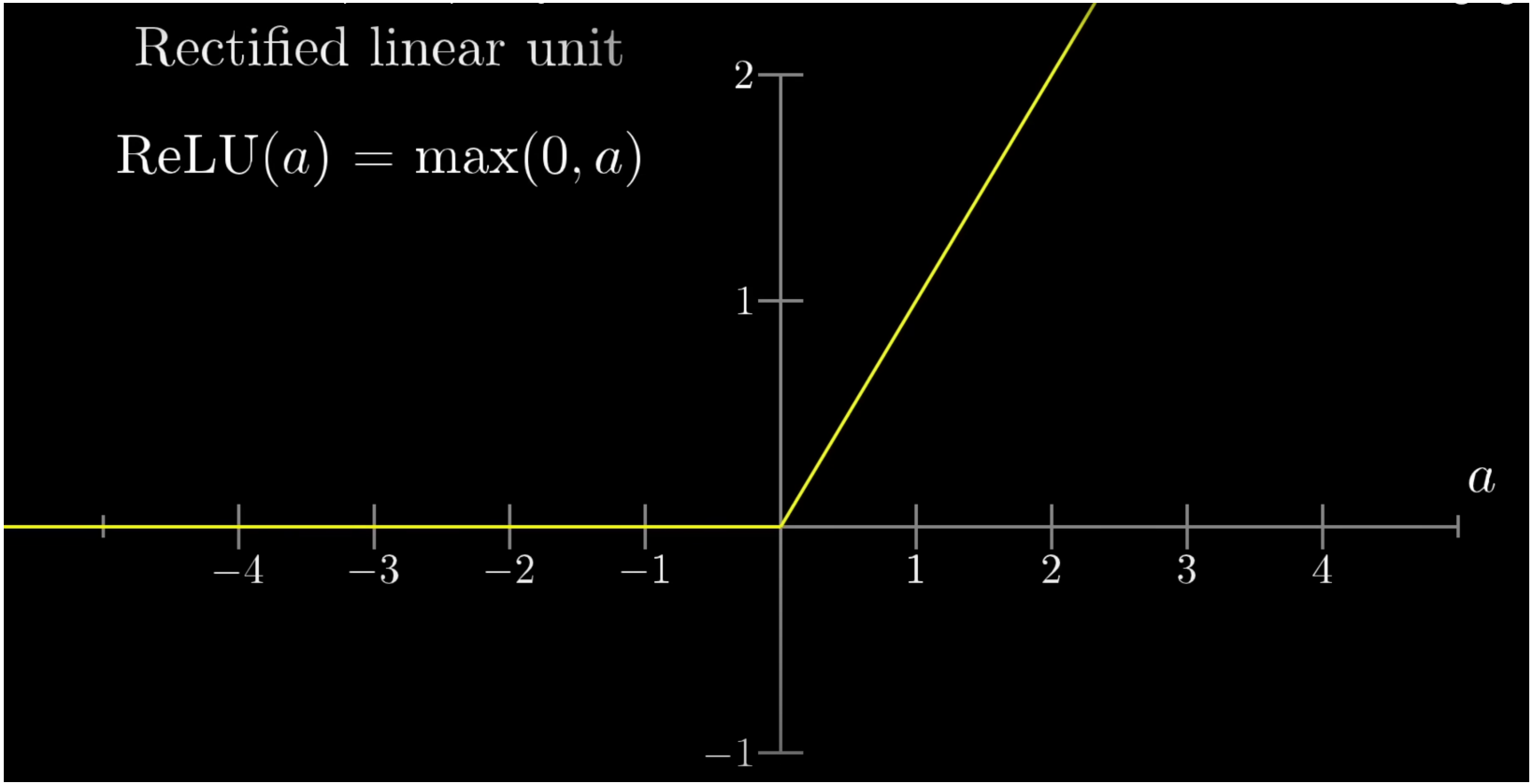
- Why is activation function useful?

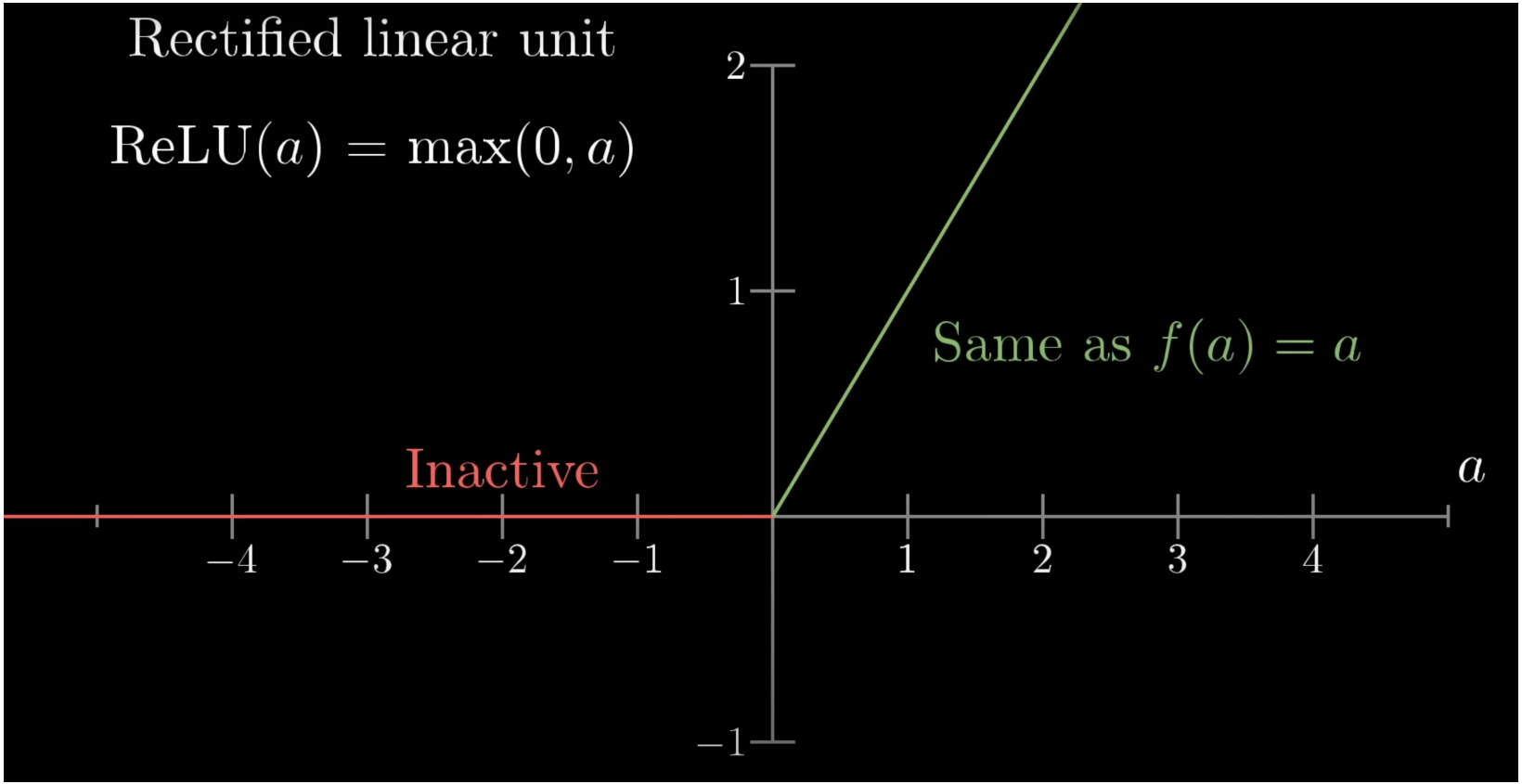
Activation function

- Why is activation function useful?
  - Non-linearity
    - *Great for deep learning!*
  - Gradient Propagation
    - *Ensure back propagation*
  - Regularization
    - *How (Implicit) Regularization of ReLU Neural Networks Characterizes the Learned Function (Jakob, 2023)*

ReLU vs. Sigmoid


Sigmoid
$\sigma(a) = \frac{1}{1+e^{-a}}$
2
1
−1
−4
−3
−2
−1
1
2
3
4
a


Rectified linear unit
$\mathrm{ReLU}(a) = \max(0, a)$
2
1
−1
a
−4
−3
−2
−1
1
2
3
4


Rectified linear unit
$\mathrm{ReLU}(a) = \max(0, a)$
Same as $f(a) = a$
Inactive
a
2
1
−1
−4
−3
−2
−1
1
2
3
4

$$Max(0,x)$$

ReLU

$$\frac{1}{1+e^{-x}}$$

Softmax

Let's see how MLP works in code