Day-85, Feb 25, 2025 (Falgun 13, 2081)

① Classification with a Neural Network: Motivation

② Classification with a Neural Network: Minimizing log-loss

③ Gradient Descent and back propagation with figures

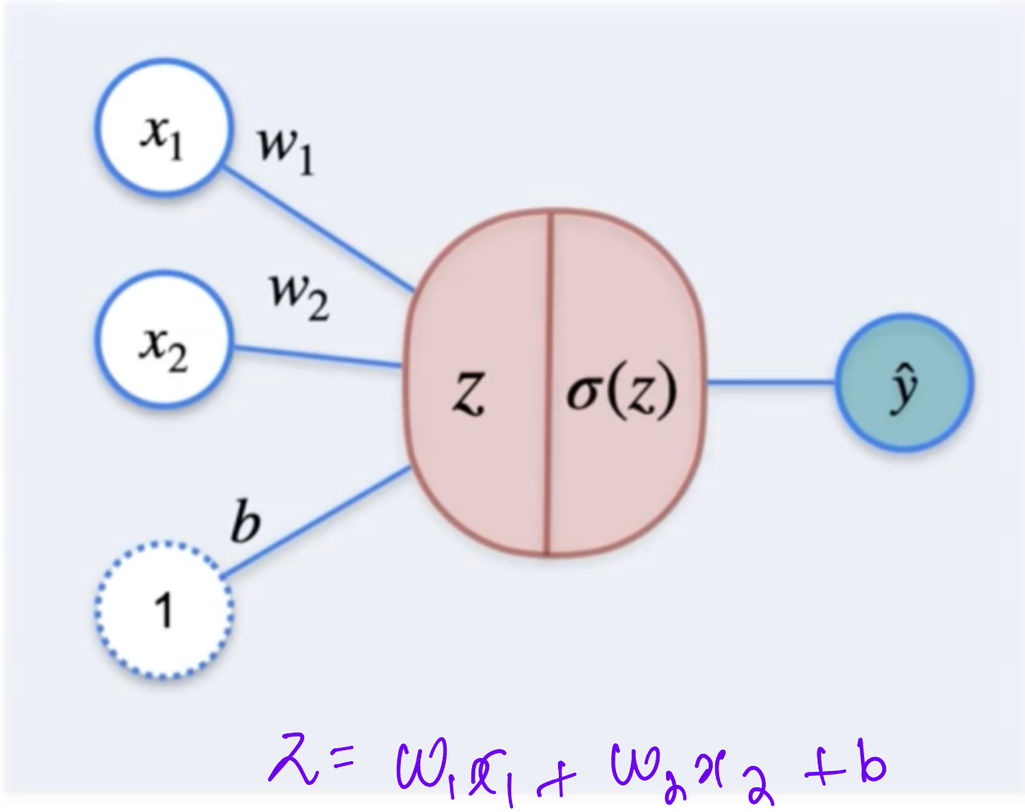
④ Newton's Method

⑤ An Example in Newton's Method.

Source: Calculus for Machine Learning and Data Science: Coursera offered by Deeplearning.AI.

Classification Problem Motivation

Sentence	Aack	Beep	Mood
Aack aack aack!	*3*	*0*	*Happy* 😊
Beep beep!	*0*	*2*	*Sad* 😔
Aack beep beep beep!	*1*	*3*	*Sad* 😔
Aack beep aack!	*2*	*1*	*Happy* 😊

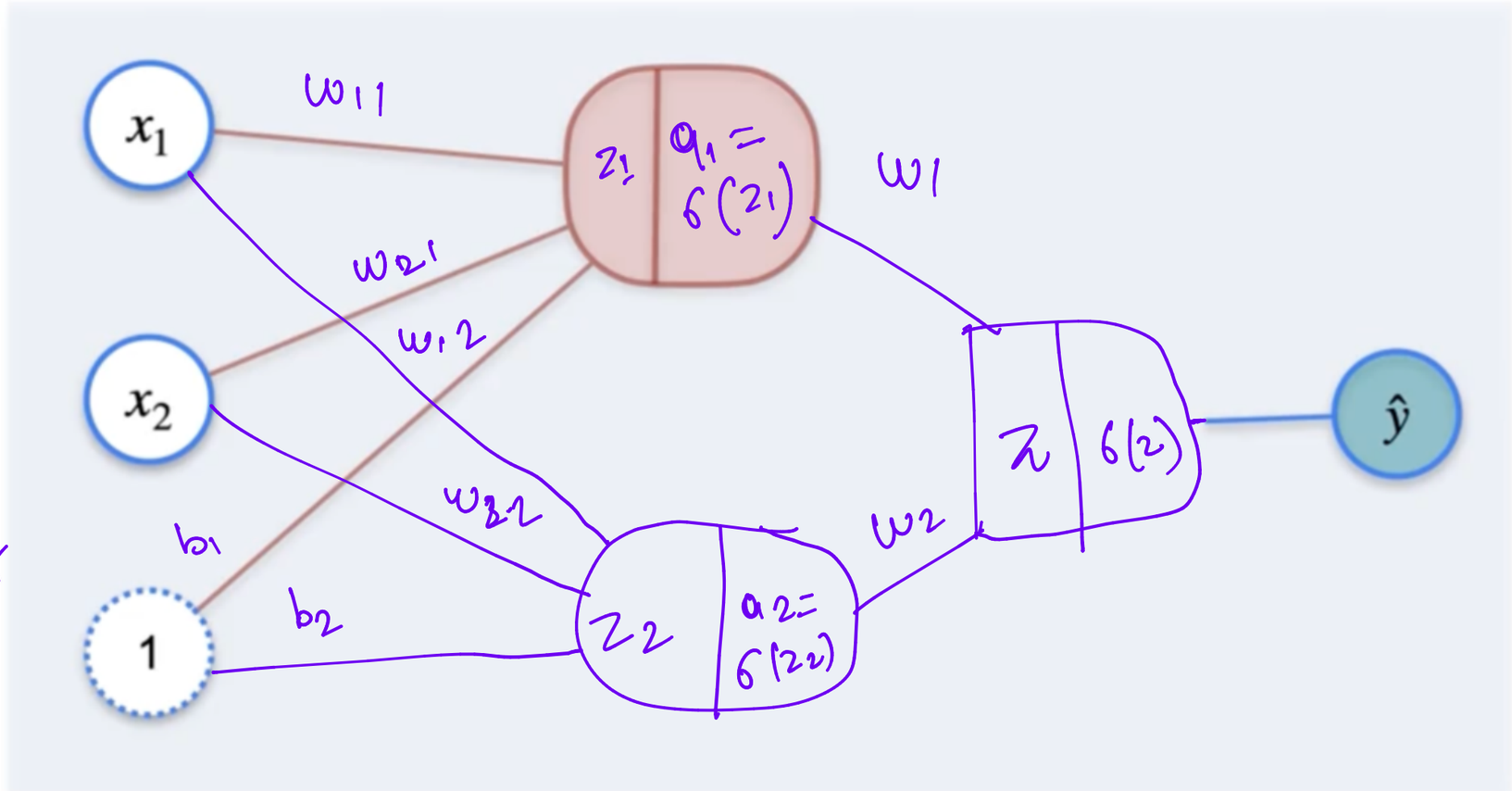


of the previous variables time
multiplied by the weights

DeepLearning.AI

2,2,1 Neural Network

- Neural network of depth 2
  - One input layer
  - One hidden layer
  - One output layer



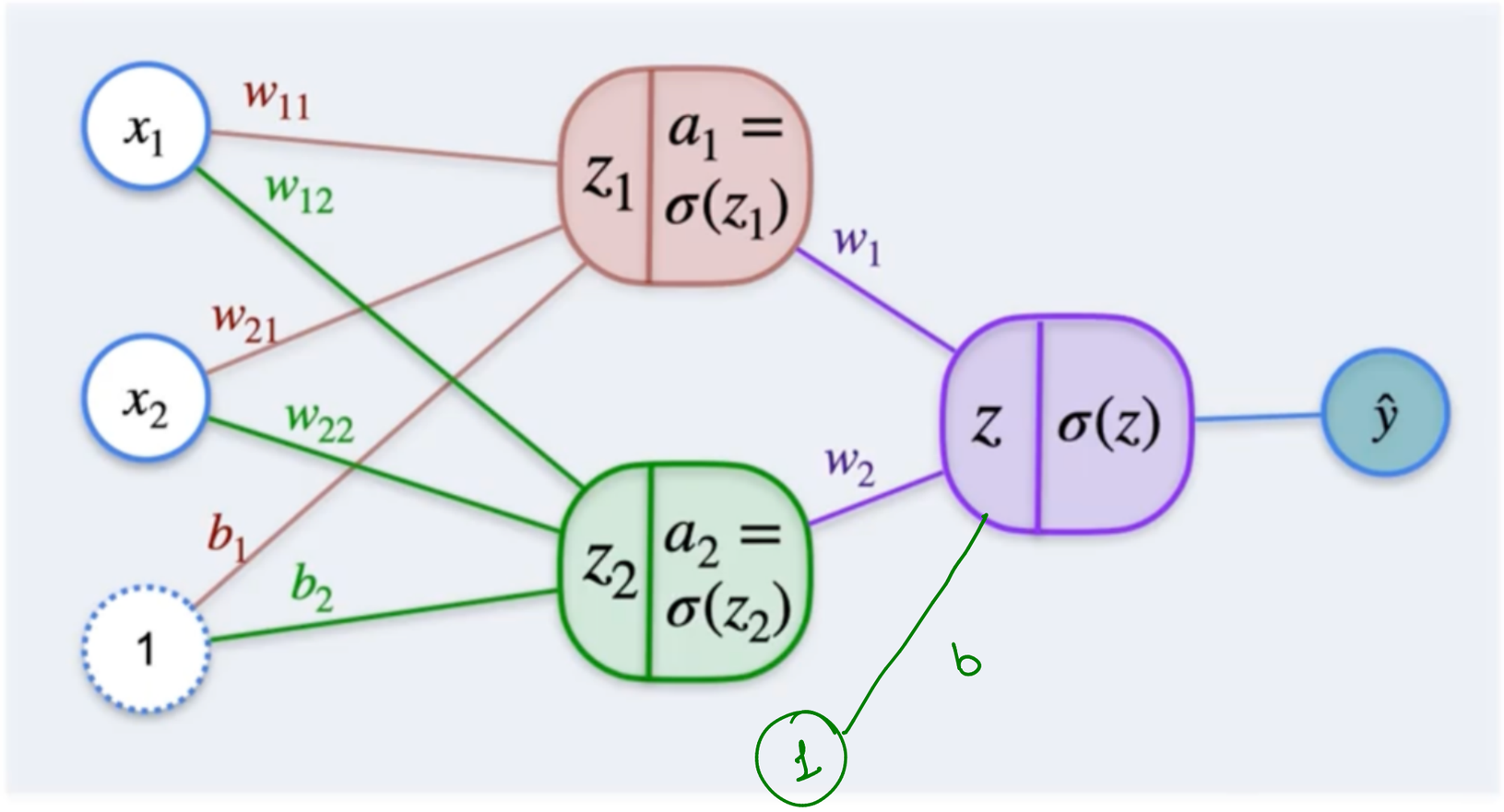
Here we have one perceptron,

DeepLearning.AI

2,2,1 Neural Network

Neural network of depth 2

- one input layer
- one hidden layer
- one output layer



Perceptrons normally have a bias

DeepLearning.AI

2,2,1 Neural Network

$a_1 = \sigma(z_1)$

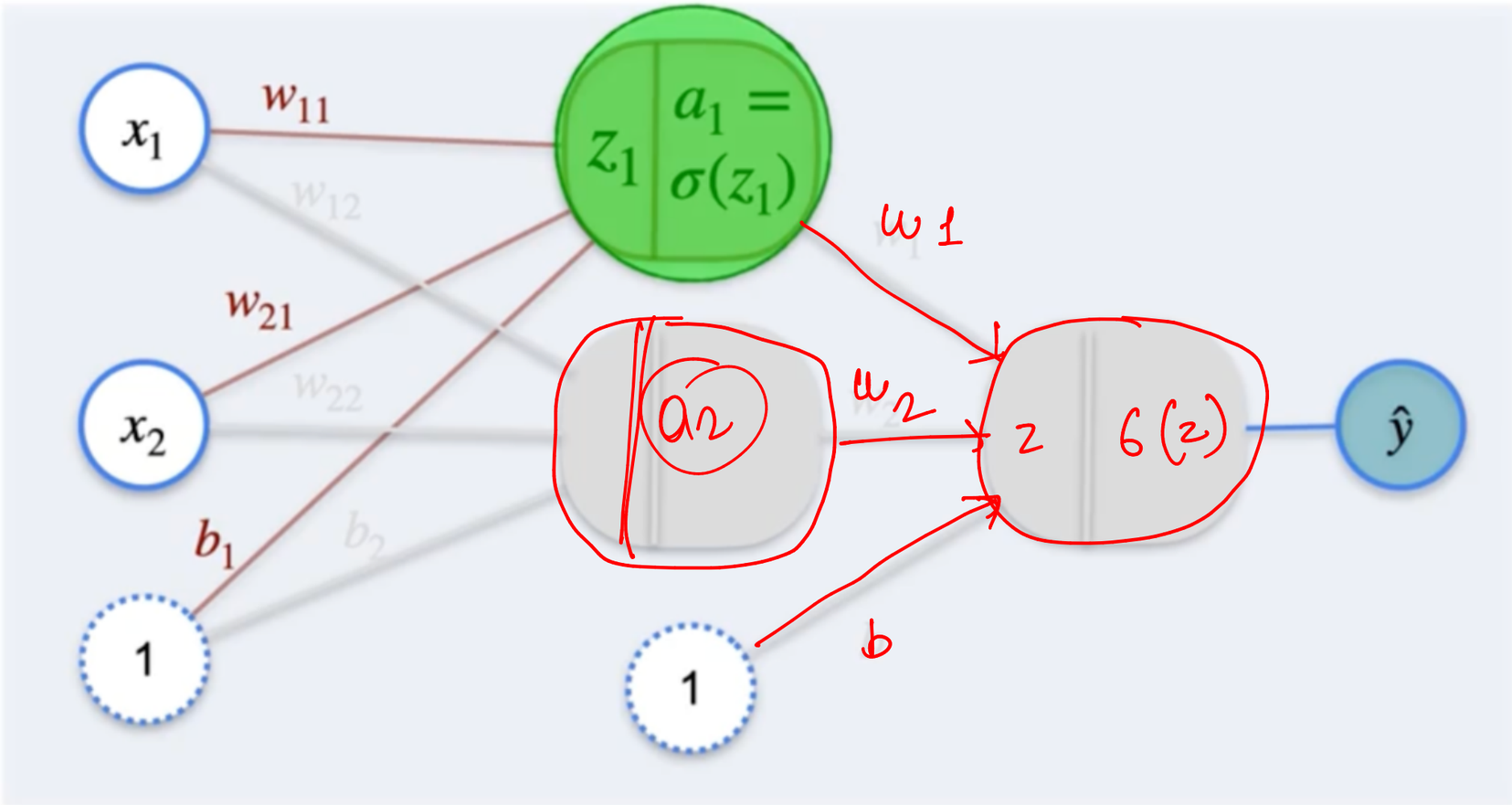
$z_1 = x_1 w_{11} + x_2 w_{21} + b_1$

$a_2 = \sigma(z_2)$

$z_2 = x_1 w_{12} + x_2 w_{22} + b_2$

$\hat{y} = \sigma(z)$

$z = a_1 w_1 + a_2 w_2 + b$



Focusing on this node alone,

2,2,1 Neural Network

$$a_1 = \sigma(z_1)$$

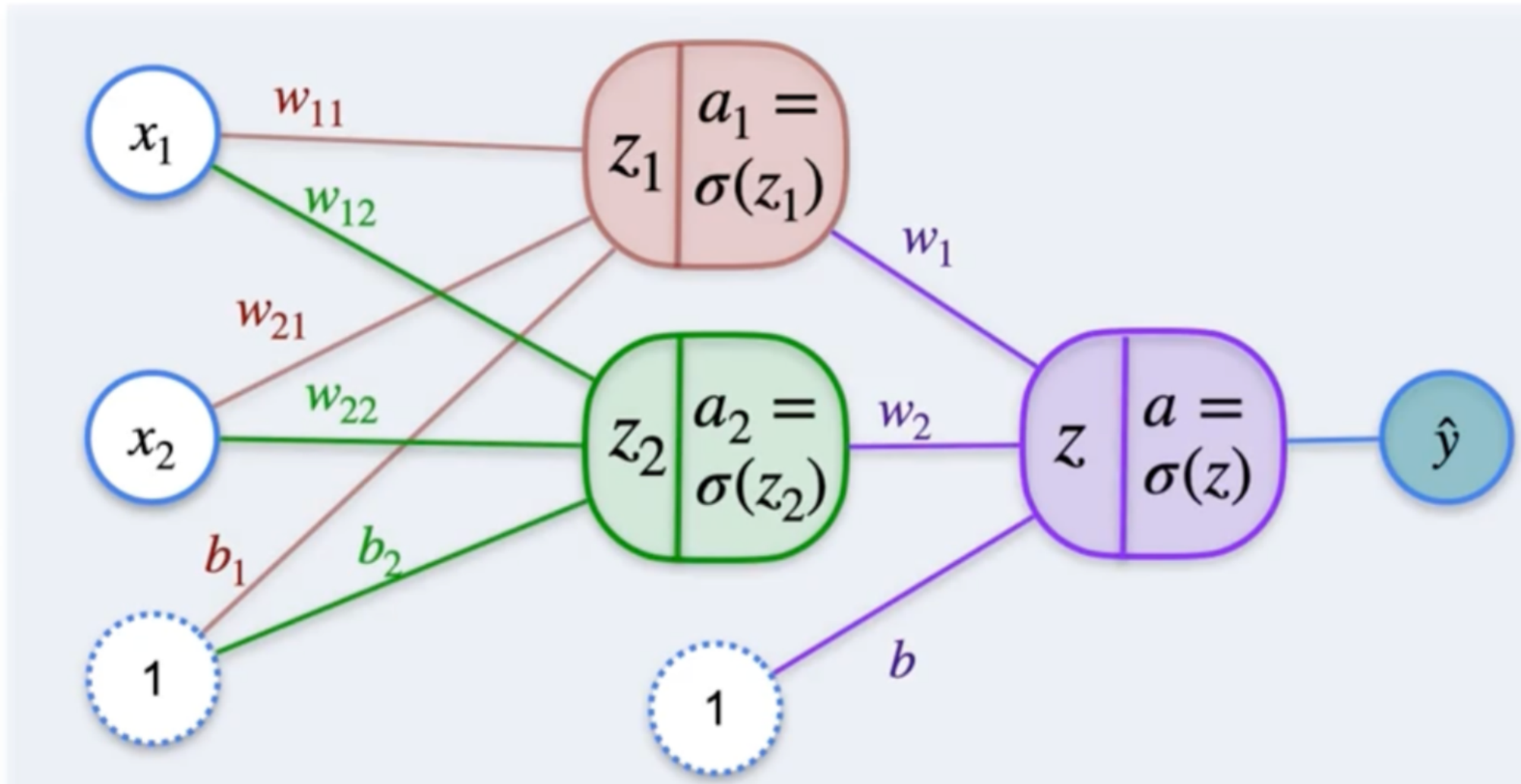
$$z_1 = x_1 w_{11} + x_2 w_{21} + b_1$$

$$a_2 = \sigma(z_2)$$

$$z_2 = x_1 w_{12} + x_2 w_{22} + b_2$$

$$\hat{y} = \sigma(z)$$

$$z = a_1 w_1 + a_2 w_2 + b$$



As a reminder, here are all the variables.

DeepLearning.AI

The values for z_1 z_1 and z_2 is:

$$z_1 = w_{11}x_1 + w_{21}x_2 + b_1$$

$$z_2 = w_{12}x_1 + w_{22}x_2 + b_2$$

$$z = w_1x_1 + w_2x_2 + b$$

Loss function

$$\rightarrow L(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$$L(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

2,2,1 Neural Network

$$a_1 = \sigma(z_1)$$

$$z_1 = x_1w_{11} + x_2w_{21} + b_1$$

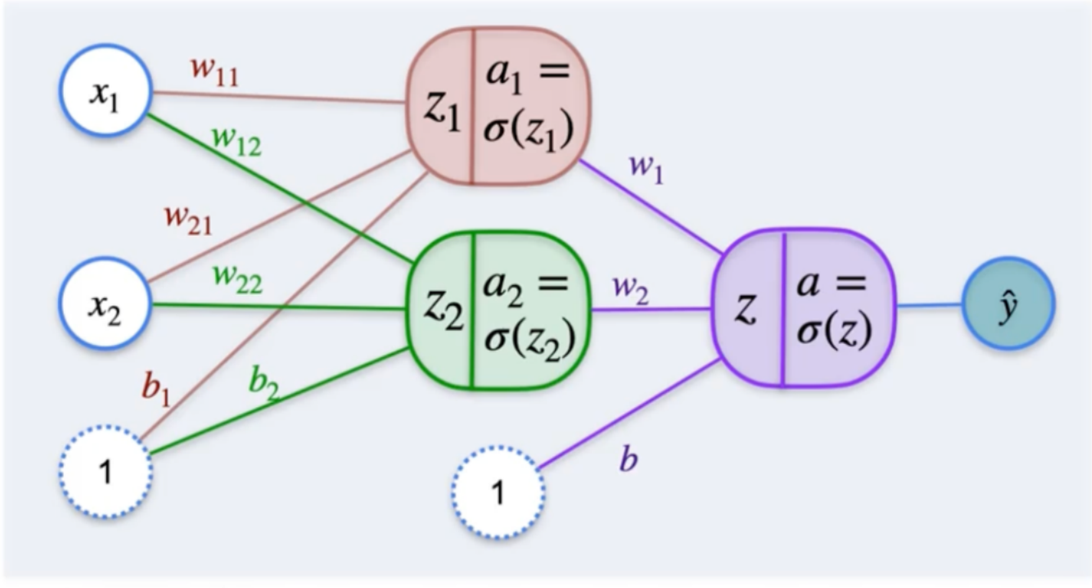
$$a_2 = \sigma(z_2)$$

$$z_2 = x_1w_{12} + x_2w_{22} + b_2$$

$$\hat{y} = \sigma(z)$$

$$z = a_1w_1 + a_2w_2 + b$$

$$L(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$



where y hat is the prediction

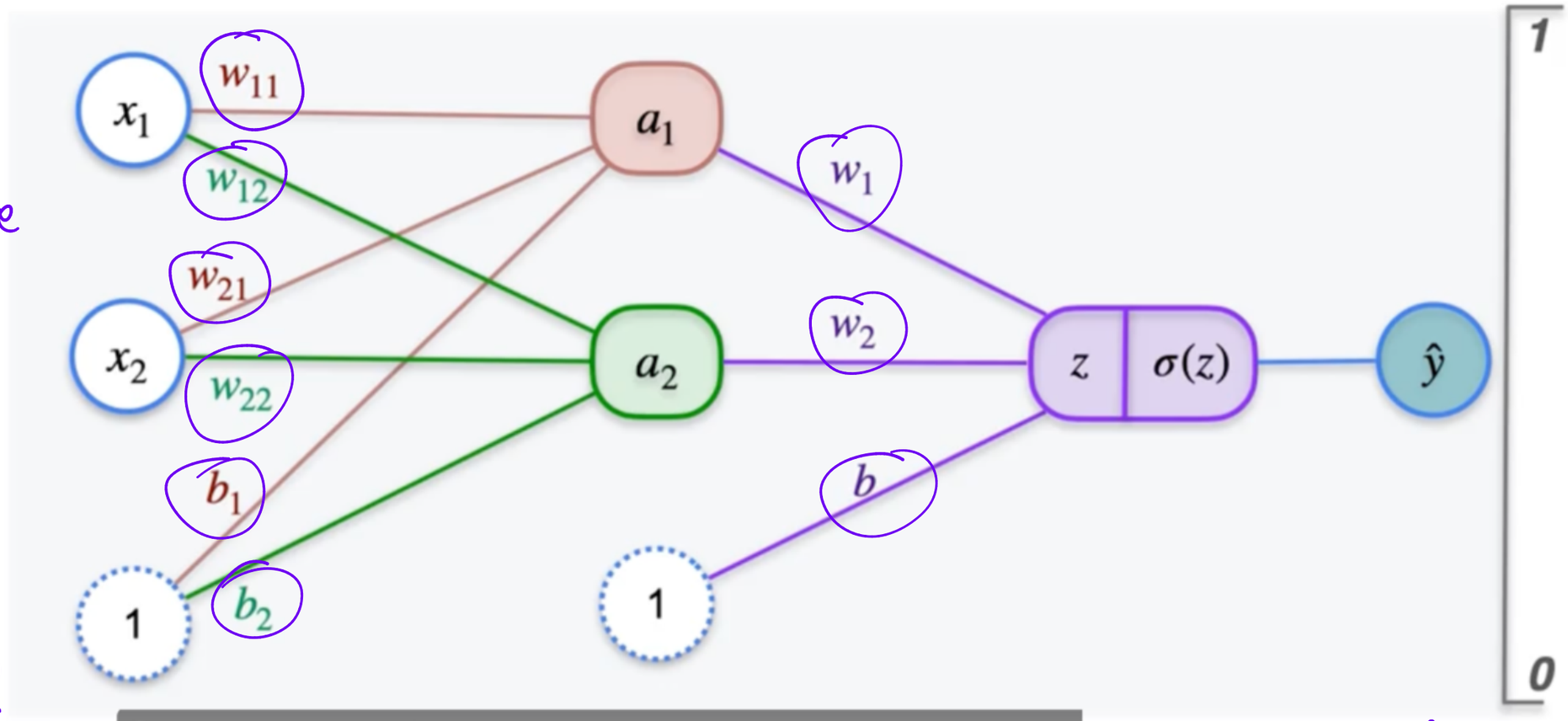
Goal:

2,2,1 Neural Network

Goal

Adjust each of the highlighted weights and biases.

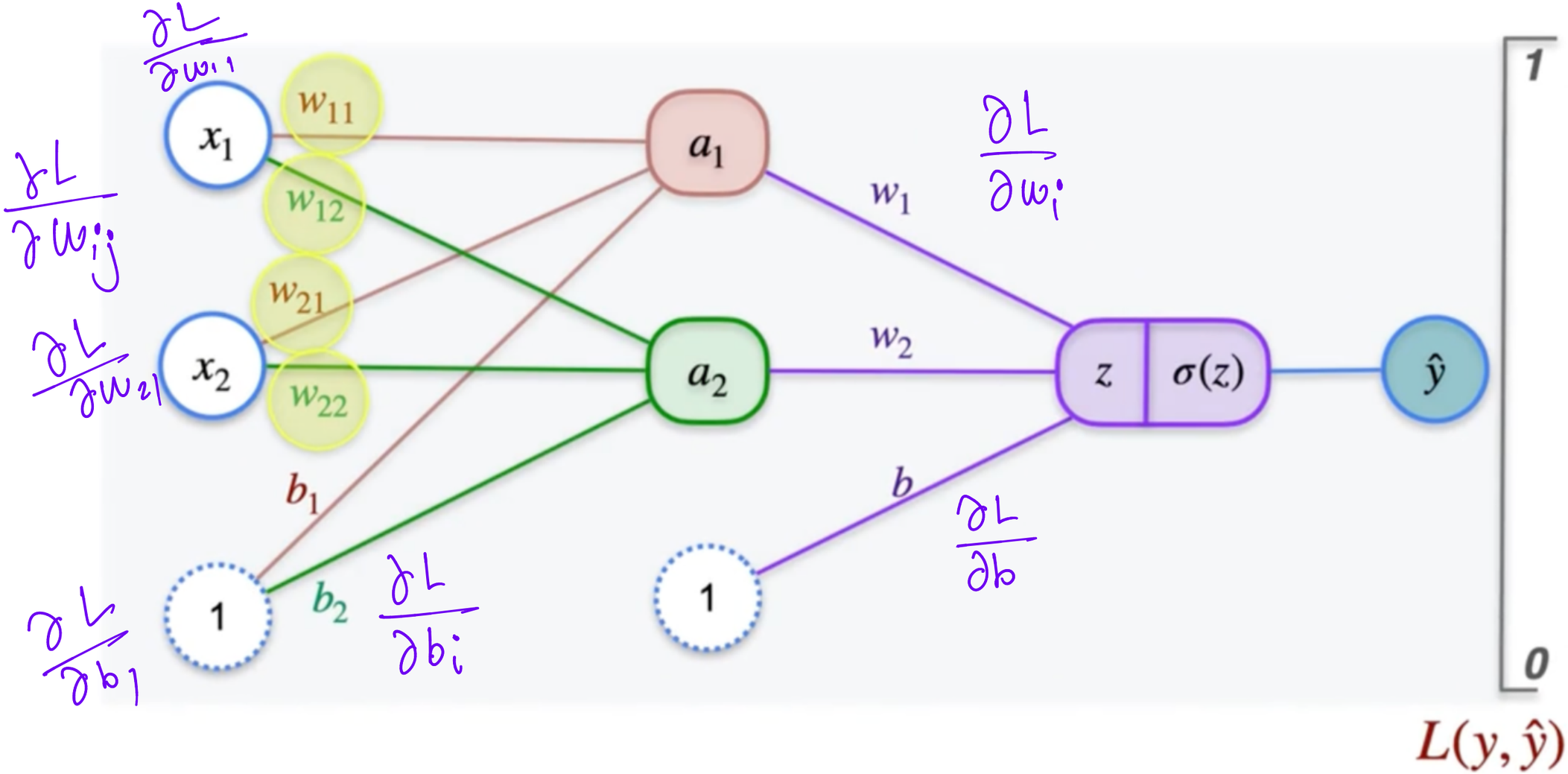
#To Reduce the loss function?



So recall that the goal is to adjust each one of the highlighted weights and

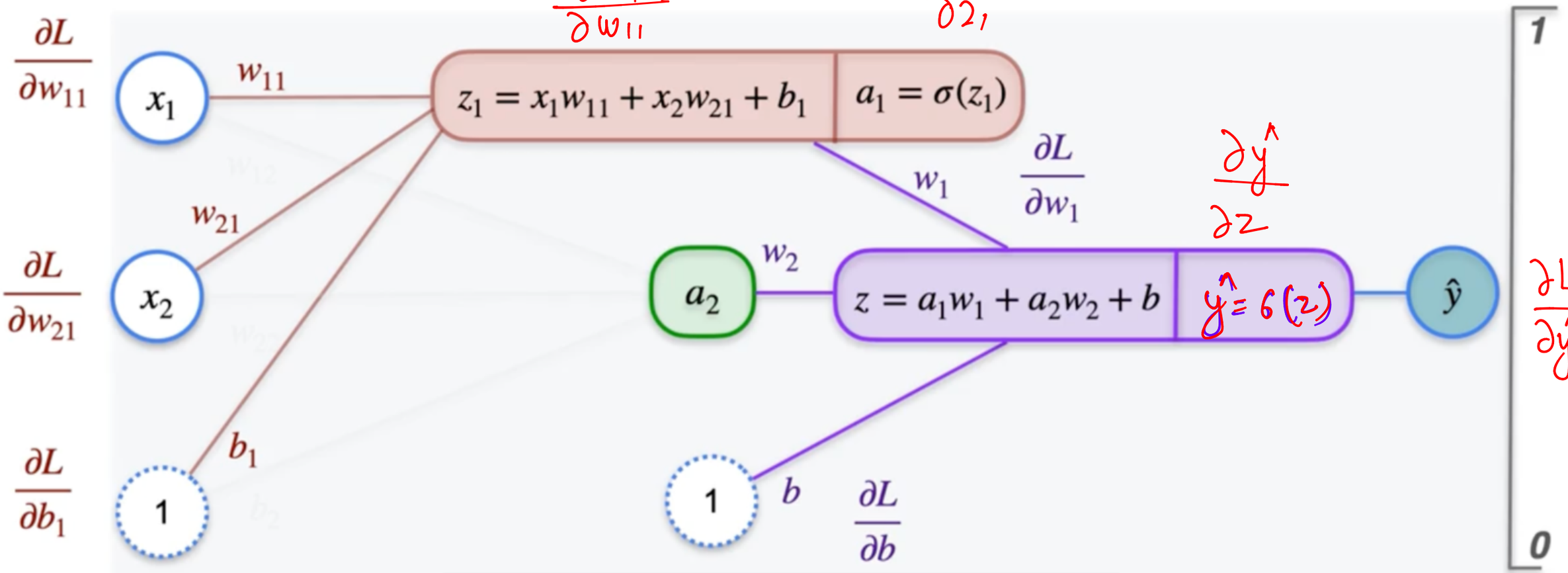
DeepLearning.AI

2,2,1 Neural Network



Reduce log loss.

2,2,1 Neural Network



a2 times the weights W1 and W2 and the bias b.

DeepLearning.AI

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial z_1}{\partial w_{11}} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z}{\partial a_1} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial L}{\partial \hat{y}}$$

2,2,1 Neural Network

$$L(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$$\hat{y} = \sigma(z)$$

$$z = a_1 w_1 + a_2 w_2 + b$$

$$a_1 = \sigma(z_1)$$

$$z_1 = x_1 w_{11} + x_2 w_{21} + b_1$$

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial z_1}{\partial w_{11}} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z}{\partial a_1} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial L}{\partial \hat{y}}$$

$$\frac{\partial L}{\partial w_{11}} = x_1 \cdot a_1(1-a_1)\, w_1\, \hat{y}(1-\hat{y}) \cdot \frac{-(y-\hat{y})}{\hat{y}(1-\hat{y})}$$

$$\left[\frac{\partial L}{\partial w_{11}} \Rightarrow -x_1 a_1 w_1 (1-a_1)(y-\hat{y})\right]$$

Perform gradient descent with

$$w_{11} \rightarrow w_{11} - \alpha \cdot x_1 \cdot w_1 a_1(1-a_1)(y-\hat{y})$$

to find optimal value of w_{11} that gives the least error.

2,2,1 Neural Network

$$L(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$$\hat{y} = \sigma(z)$$

$$z = a_1 w_1 + a_2 w_2 + b$$

$$a_1 = \sigma(z_1)$$

$$z_1 = x_1 w_{11} + x_2 w_{21} + b_1$$

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial z_1}{\partial w_{11}} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z}{\partial a_1} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial L}{\partial \hat{y}}$$

$$\frac{\partial L}{\partial w_{11}} = x_1 \cdot a_1(1-a_1) \cdot w_1 \cdot \cancel{\hat{y}(1-\hat{y})} \cdot \frac{-(y - \hat{y})}{\cancel{\hat{y}(1-\hat{y})}}$$

$$= -x_1 w_1 a_1 (1-a_1)(y - \hat{y})$$

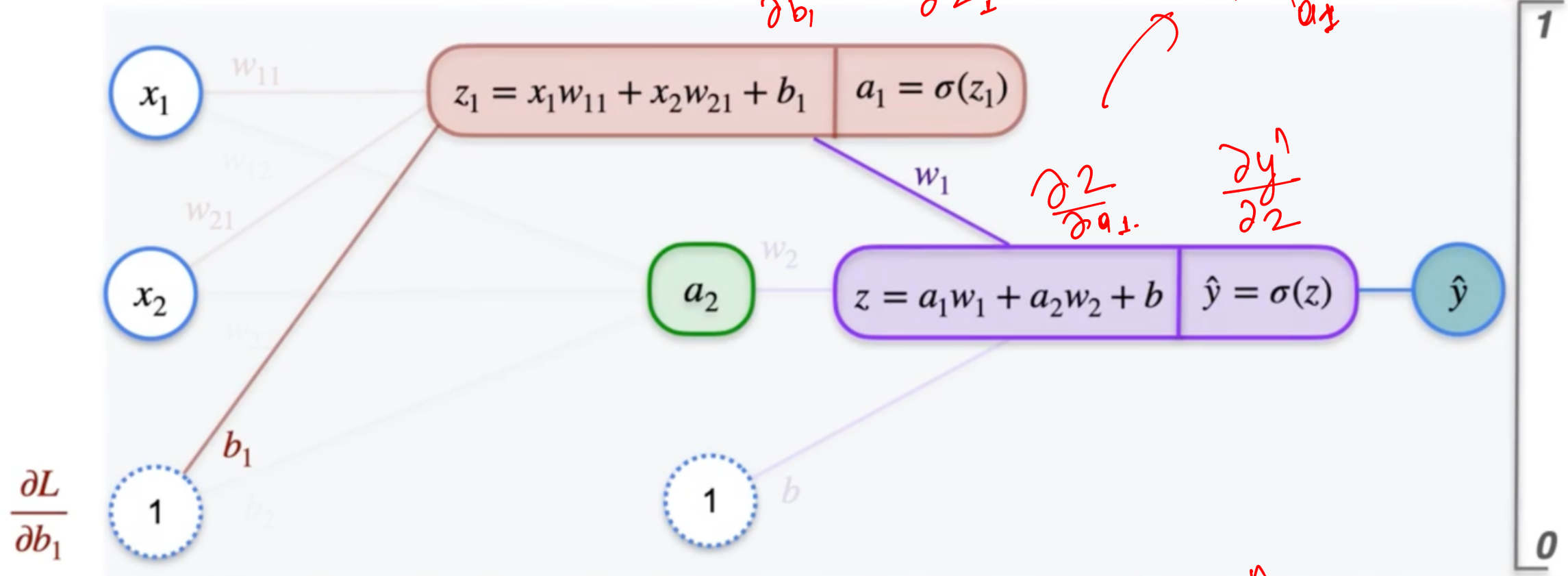
Perform gradient descent with

$$w_{11} \rightarrow w_{11} - \alpha \text{-} x_1 w_1 a_1 (1-a_1)(y - \hat{y})$$

to find optimal value of w_{11} that gives the least error

DeepLearning.AI

2,2,1 Neural Network



$$\frac{\partial L}{\partial b_1} \Rightarrow \frac{\partial z_1}{\partial b_1} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z}{\partial a_1} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial L}{\partial \hat{y}}$$

DeepLearning.AI

2,2,1 Neural Network

$$L(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$$\hat{y} = \sigma(z)$$

$$z = a_1 w_1 + a_2 w_2 + b$$

$$a_1 = \sigma(z_1)$$

$$z_1 = x_1 w_{11} + x_2 w_{21} + b_1$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial z_1}{\partial b_1} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z}{\partial a_1} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial L}{\partial \hat{y}}$$

$$\frac{\partial L}{\partial b_1} = 1 \cdot a_1(1-a_1)\, w_1\, \hat{y}(1-\hat{y}) \cdot \frac{-(y-\hat{y})}{\hat{y}(1-\hat{y})}$$

Sigmoid f'

$$\Rightarrow -w_1 a_1 (1-a_1)(y-\hat{y})$$

Perform Gradient Descent with

$$b_1 \rightarrow b_1 - \alpha\left(-w_1 a_1 (1-a_1)(y-\hat{y})\right)$$

DeepLearning.AI

2,2,1 Neural Network

$$L(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$$\hat{y} = \sigma(z)$$

$$z = a_1 w_1 + a_2 w_2 + b$$

$$a_1 = \sigma(z_1)$$

$$z_1 = x_1 w_{11} + x_2 w_{21} + b_1$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial z_1}{\partial b_1} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z}{\partial a_1} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial L}{\partial \hat{y}}$$

$$\frac{\partial L}{\partial b_1} = 1 \cdot a_1(1-a_1) \cdot w_1 \cdot \cancel{\hat{y}(1-\hat{y})} \cdot \frac{-(y - \hat{y})}{\cancel{\hat{y}(1-\hat{y})}}$$

$$= -w_1 a_1 (1-a_1)(y - \hat{y})$$

Perform gradient descent with

$$b_1 \rightarrow b_1 - \alpha(-w_1 a_1 (1-a_1)(y - \hat{y}))$$

to find optimal value of b_1 that gives the least error

DeepLearning.AI

2,2,1 Neural Network

→ Weight updates.

$$w_{11} \rightarrow w_{11} + \alpha x_1 w_1 a_1 (1 - a_1)(y - \hat{y})$$

$$w_{12} \rightarrow w_{11} + \alpha x_2 w_1 a_1 (1 - a_1)(y - \hat{y})$$

$$b_1 \rightarrow b_1 + \alpha w_1 a_1 (1 - a_1)(y - \hat{y})$$

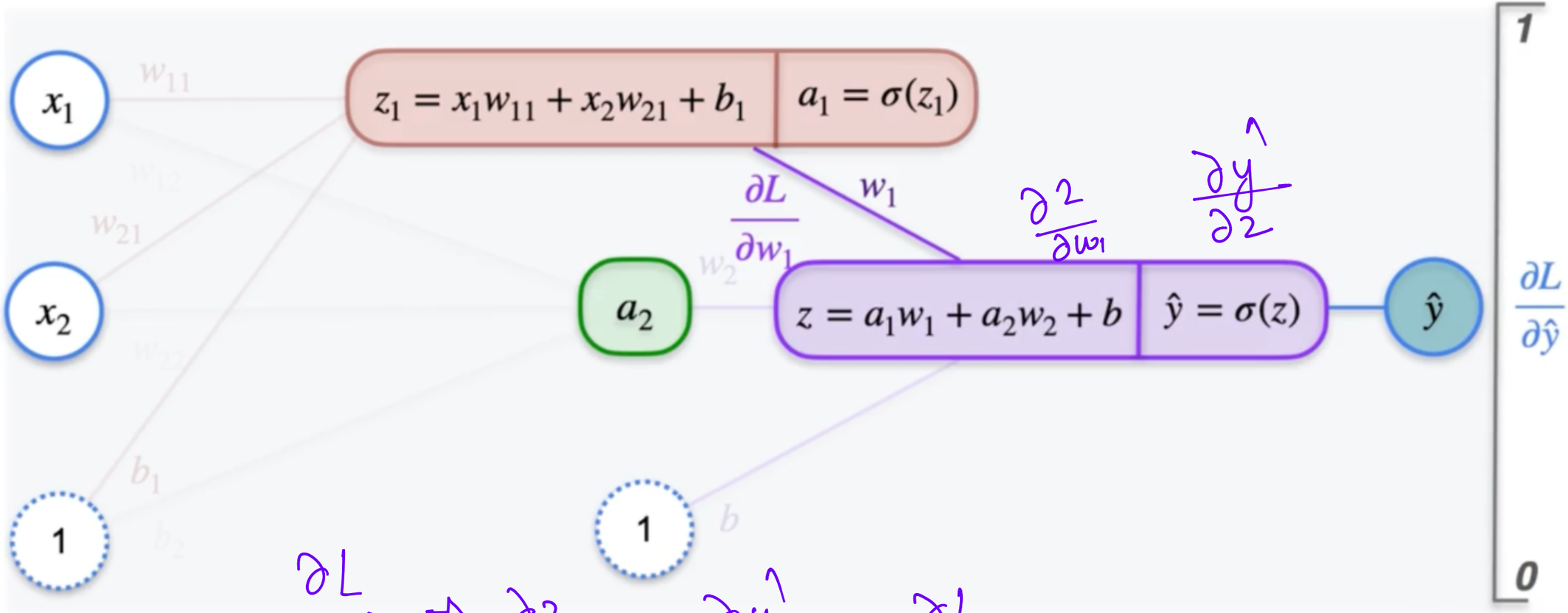
$$w_{21} \rightarrow w_{21} + \alpha x_1 w_2 a_2 (1 - a_2)(y - \hat{y})$$

$$w_{22} \rightarrow w_{22} + \alpha x_2 w_2 a_2 (1 - a_2)(y - \hat{y})$$

$$b_2 \rightarrow b_2 + \alpha w_2 a_2 (1 - a_2)(y - \hat{y})$$

DeepLearning.AI

2,2,1 Neural Network



$$\frac{\partial L}{\partial w_1} \Rightarrow \frac{\partial z}{\partial w_1} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial L}{\partial \hat{y}}$$

$$\frac{\partial L}{\partial w_1} \Rightarrow a_1 \cdot \hat{y}(1-\hat{y}) \cdot \frac{-(y-\hat{y})}{\hat{y}(1-\hat{y})}$$

DeepLearning.AI

2,2,1 Neural Network

$$L(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$$\hat{y} = \sigma(z)$$

$$z = a_1 w_1 + a_2 w_2 + b$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial z}{\partial w_1} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial L}{\partial \hat{y}}$$

$$\frac{\partial L}{\partial w_1} = a_1 \cdot \cancel{\hat{y}(1-\hat{y})} \cdot \frac{-(y - \hat{y})}{\cancel{\hat{y}(1-\hat{y})}}$$

$$= -a_1(y - \hat{y})$$

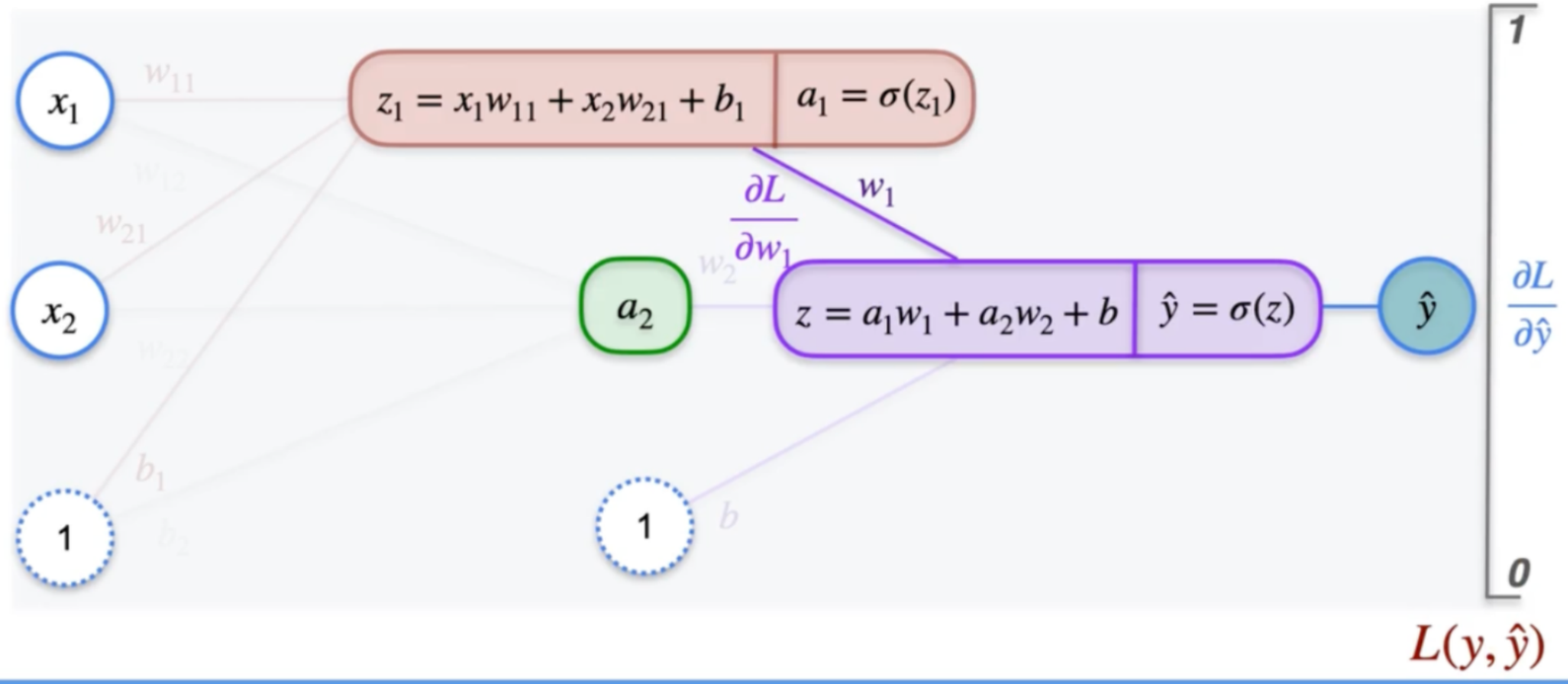
Perform gradient descent with

$$w_1 \rightarrow w_1 - \alpha(-a_1(y - \hat{y}))$$

to find optimal value of w_1 that gives the least error

DeepLearning.AI

2,2,1 Neural Network



#.

2,1 Neural Network

$$w_1 \rightarrow w_1 + \alpha a_1(y - \hat{y}))$$

$$w_2 \rightarrow w_2 + \alpha a_2(y - \hat{y}))$$

$$b \rightarrow b + \alpha(y - \hat{y}))$$

2,2,1 Neural Network

$$L(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$$\hat{y} = \sigma(z)$$

$$z = a_1w_1 + a_2w_2 + b$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial z}{\partial w_1} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial L}{\partial \hat{y}}$$

$$\frac{\partial L}{\partial w_1} = a_1 \cdot \cancel{\hat{y}(1-\hat{y})} \cdot \frac{-(y - \hat{y})}{\cancel{\hat{y}(1-\hat{y})}}$$

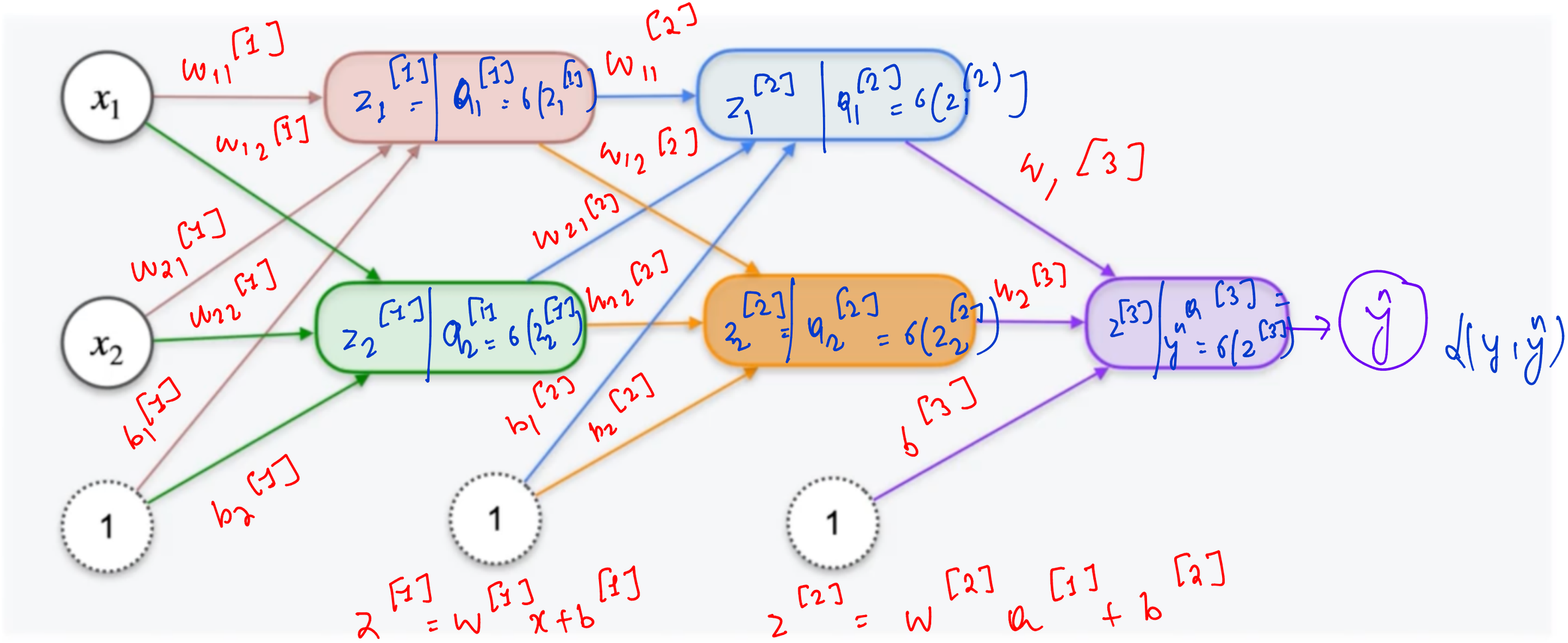
$$= -a_1(y - \hat{y})$$

Perform gradient descent with

$$w_1 \rightarrow w_1 - \alpha(-a_1(y - \hat{y}))$$

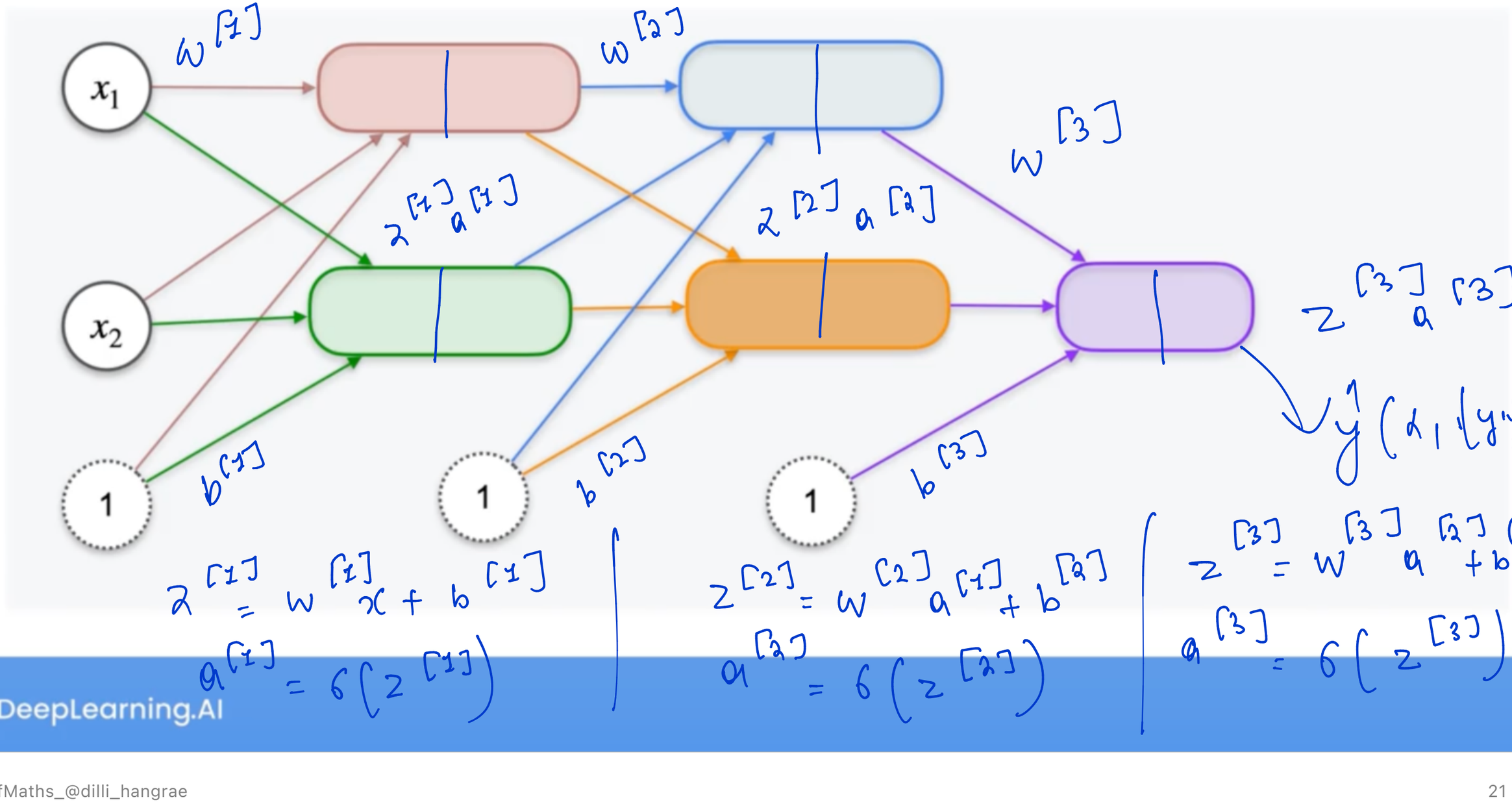
to find optimal value of w_1 that gives the least error

Back Propagation Introduction



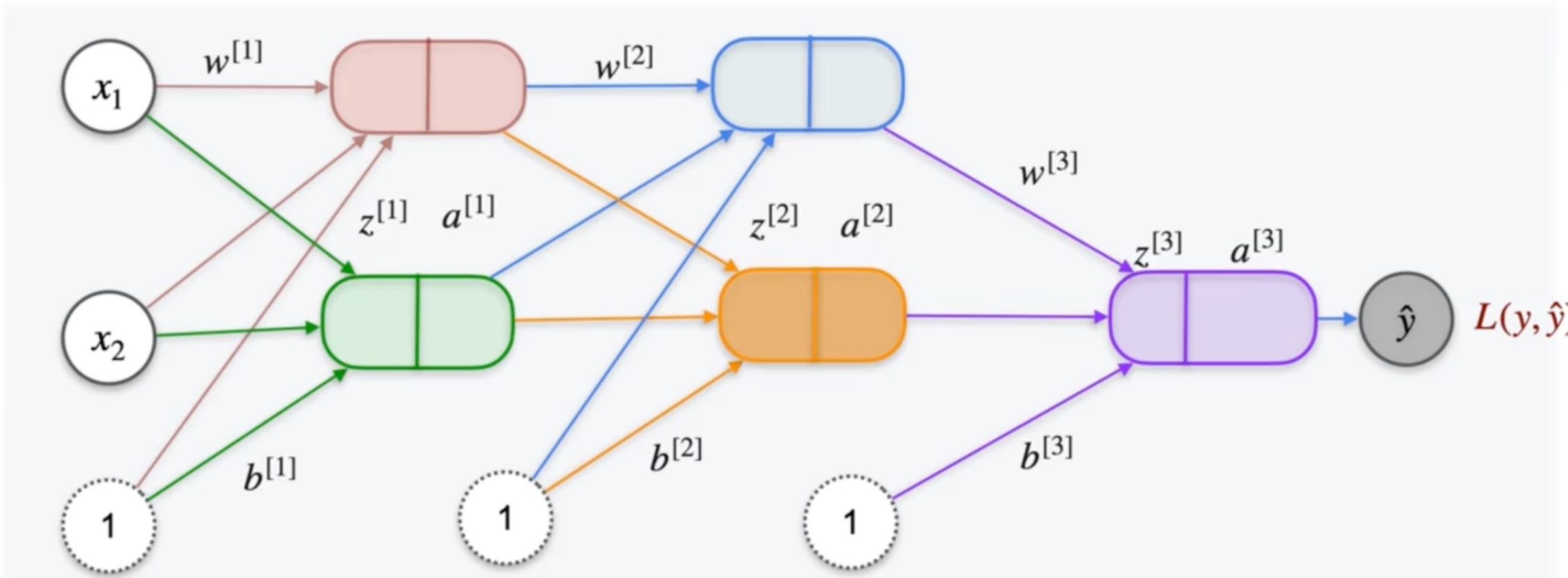
$$L(y, \hat{y}) = -y \log(\hat{y}) - (1-y) \log(1-\hat{y})$$

Back Propagation Introduction



DeepLearning.AI

Back Propagation Introduction



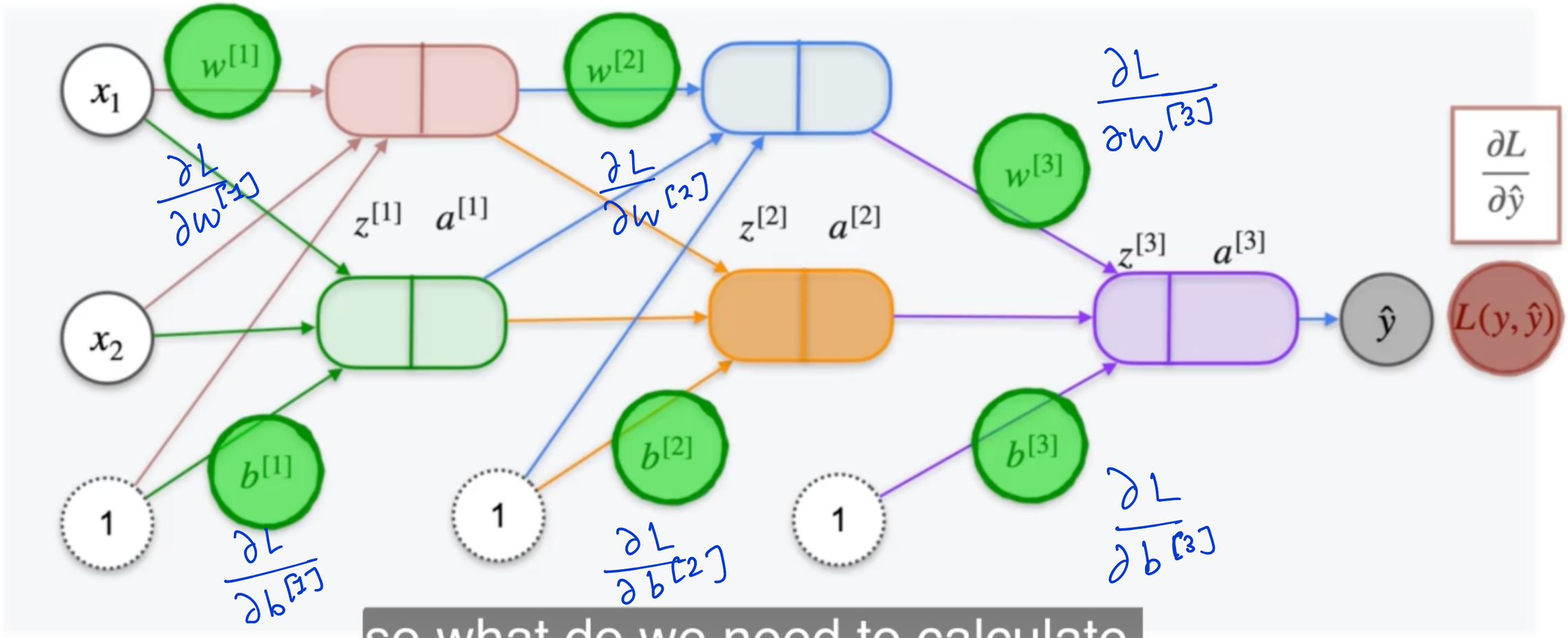
$$z^{[1]} = w^{[1]}x + b^{[1]}$$
$$a^{[1]} = \sigma(z^{[1]})$$

$$z^{[2]} = w^{[2]}a^{[1]} + b^{[2]}$$
$$a^{[2]} = \sigma(z^{[2]})$$

$$z^{[3]} = w^{[3]}a^{[2]} + b^{[3]}$$
$$a^{[3]} = \sigma(z^{[3]})$$

DeepLearning.AI

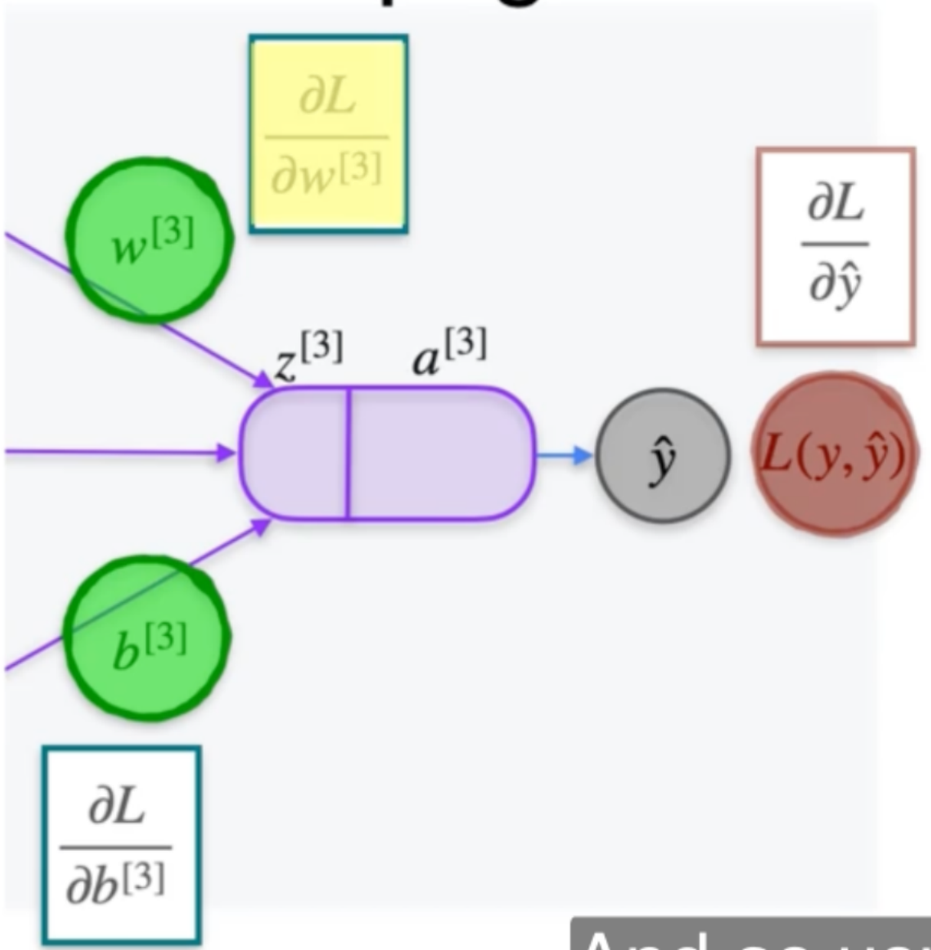
Back Propagation Introduction



so what do we need to calculate,
we need to calculate DL over DY hat and

DeepLearning.AI

Back Propagation Introduction



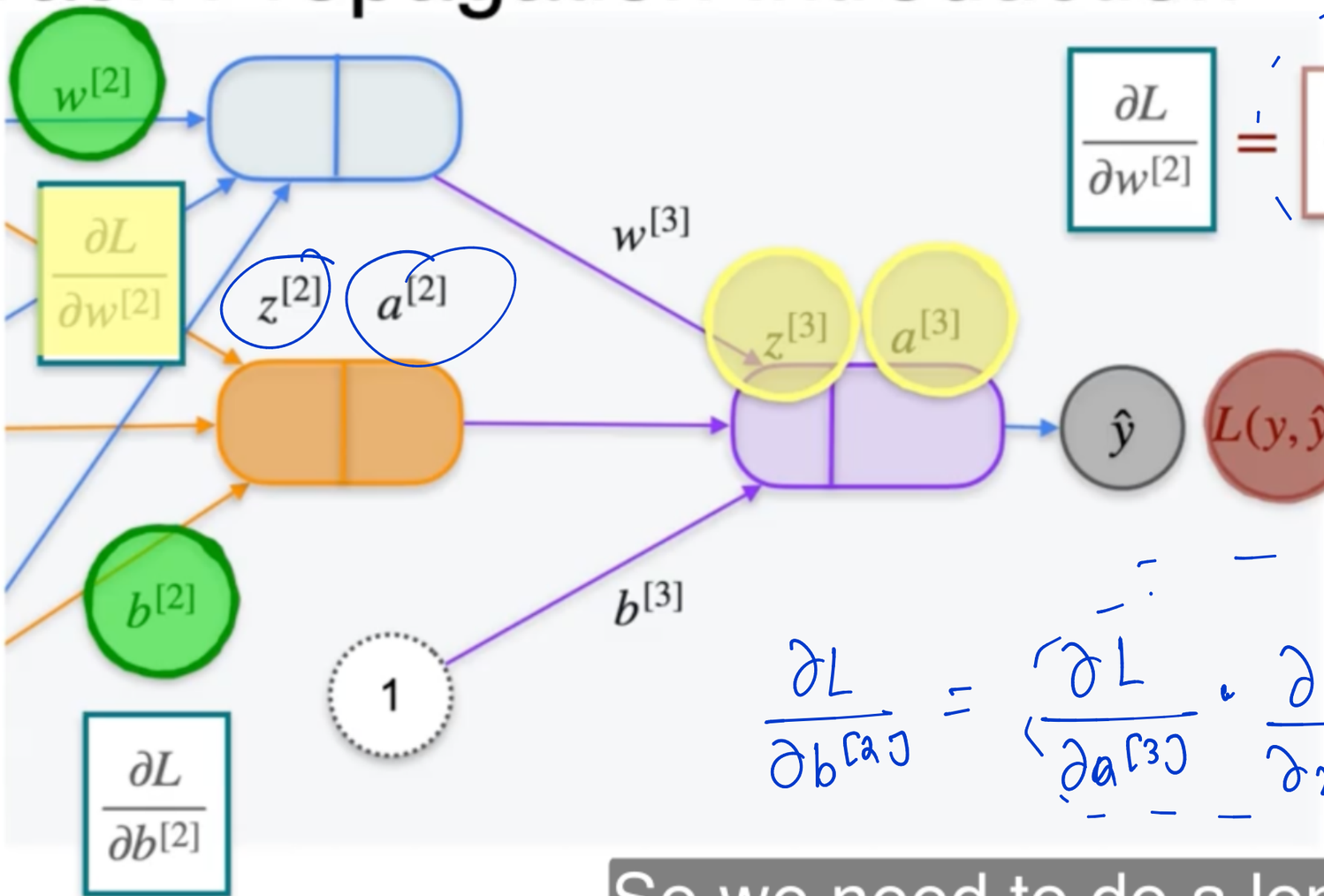
$$\frac{\partial L}{\partial w^{[3]}} = \frac{\partial z^{[3]}}{\partial w^{[3]}} \cdot \frac{\partial a^{[3]}}{\partial z^{[3]}} \cdot \frac{\partial L}{\partial a^{[3]}}$$

$$\frac{\partial L}{\partial b^{[3]}} = \frac{\partial z^{[3]}}{\partial b^{[3]}} \cdot \frac{\partial a^{[3]}}{\partial z^{[3]}} \cdot \frac{\partial L}{\partial a^{[3]}}$$

And as usual we need to build a chain because we need

DeepLearning.AI

Back Propagation Introduction



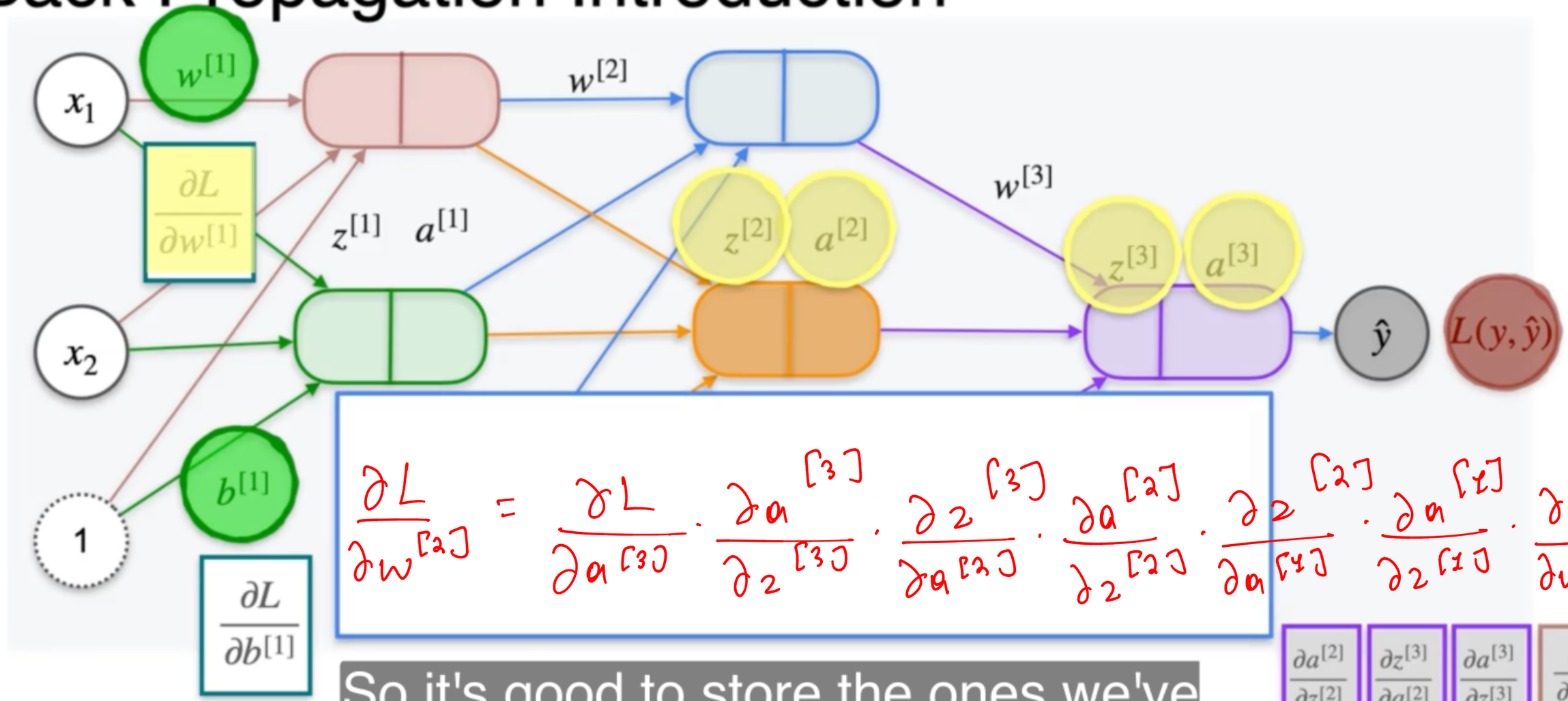
$$\frac{\partial L}{\partial w^{[2]}} = \frac{\partial L}{\partial a^{[3]}} \cdot \frac{\partial a^{[3]}}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial a^{[2]}} \cdot \frac{\partial a^{[2]}}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial w^{[2]}}$$

$$\frac{\partial L}{\partial b^{[2]}} = \left(\frac{\partial L}{\partial a^{[3]}} \cdot \frac{\partial a^{[3]}}{\partial z^{[3]}} \right) \cdot \frac{\partial z^{[3]}}{\partial a^{[2]}} \cdot \frac{\partial a^{[2]}}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial b^{[2]}}$$

So we need to do a long,
long chain rule of variables,

DeepLearning.AI

Back Propagation Introduction



So it's good to store the ones we've already calculated and finally for

DeepLearning.AI

Newton's Method (Alternate Method of Gradient Descent):

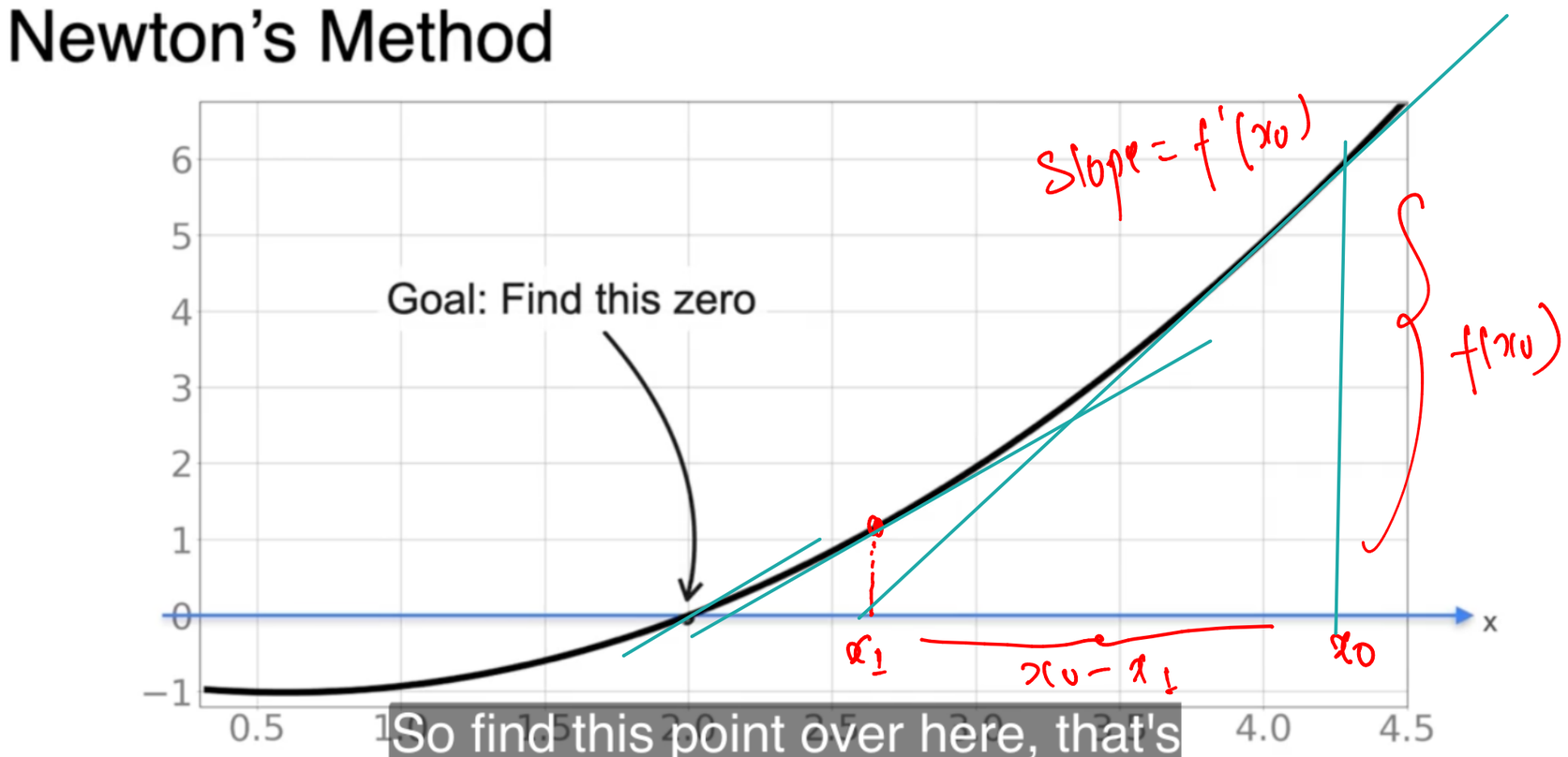
$$\frac{f(x_0)}{x_0 - x_1} = f'(x_0)$$

$$\frac{\text{rise}}{\text{run}} = \text{Slope}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Also

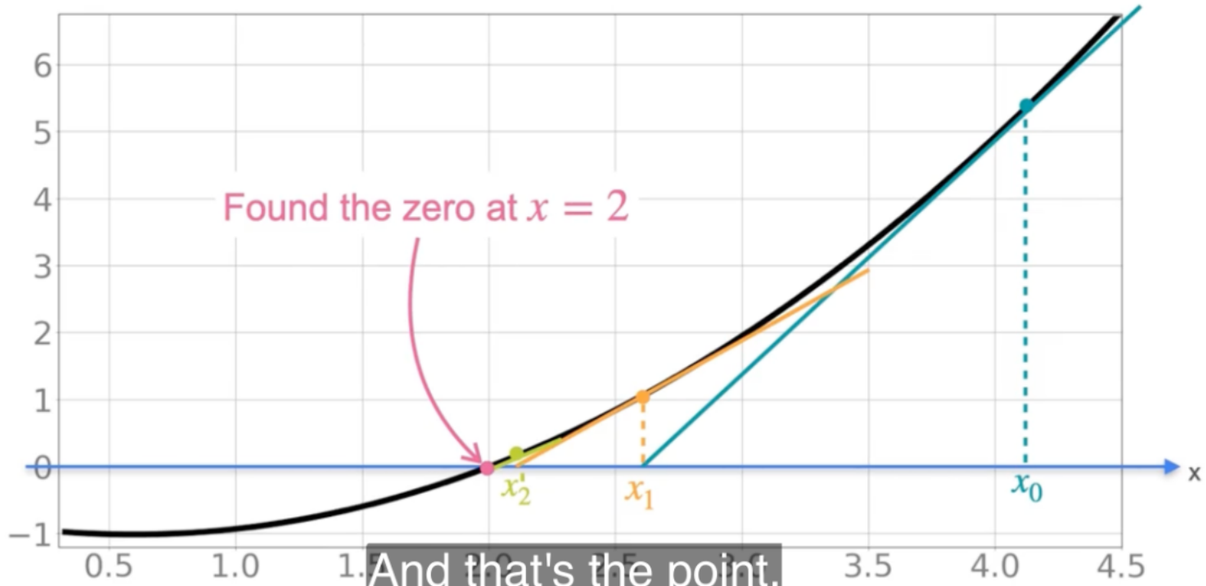
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$



So find this point over here, that's the point where F of X is equal to zero.

DeepLearning.AI

Newton's Method



And that's the point,
to be able to approximate the zero,

DeepLearning.AI

Newton's Method for Optimization.

Goal: find a zero of $f(x)$.

minimize $g(x)$ → find zeros of $g'(x)$

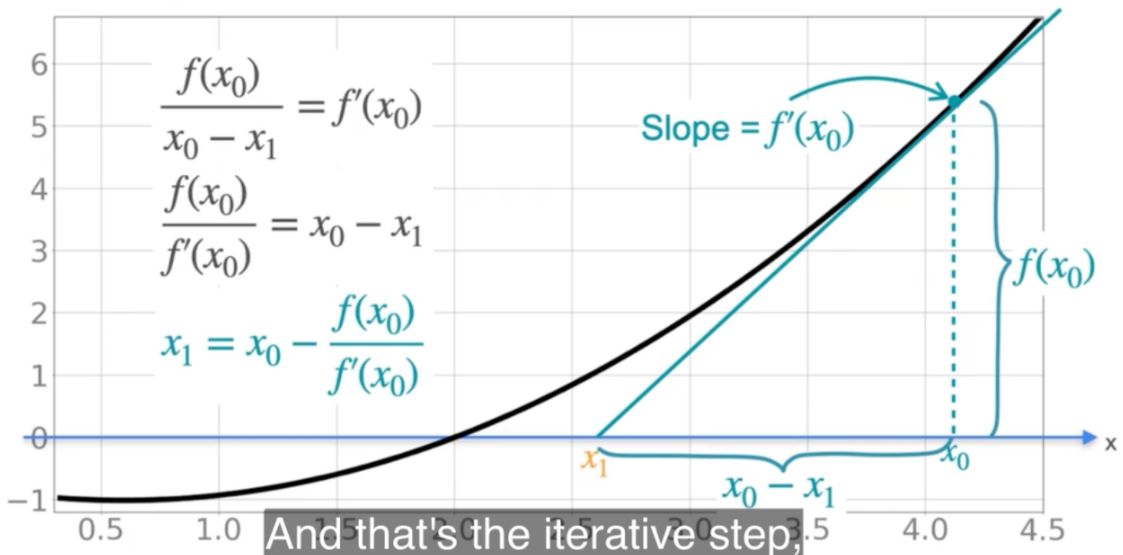
$f(x) \mapsto g'(x)$ $\quad f'(x) \mapsto (g'(x))'$

1) Start with some x_0.

2) Update:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Update Approximation

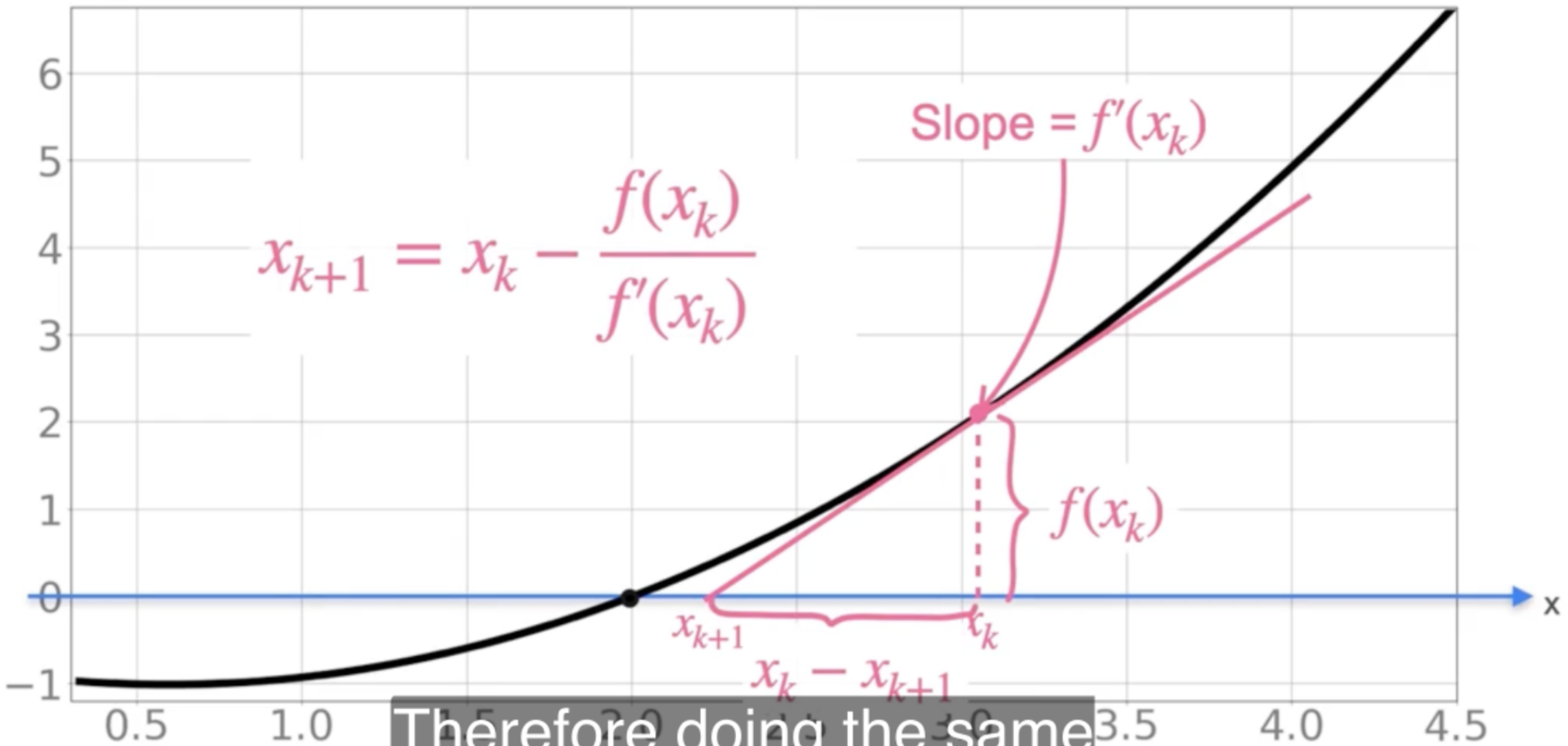


And that's the iterative step,
you have F zero,

DeepLearning.AI

Update Approximation

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$



Therefore doing the same methods before escape plus one

DeepLearning.AI

Summarize Algorithm:

Newton's Method for Optimization

Newton's method

Goal: find a zero of $f(x)$

1) Start with some x_0

2) Update:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

3) Repeat 2) until you find the root.

NM for Optimization

Goal: minimize $g(x)$ ⇒ find zeros of $g'(x)$

$f(x) \mapsto g'(x)$ $\quad$ $f'(x) \mapsto (g'(x))'$

1) Start with some x_0

2) Update:

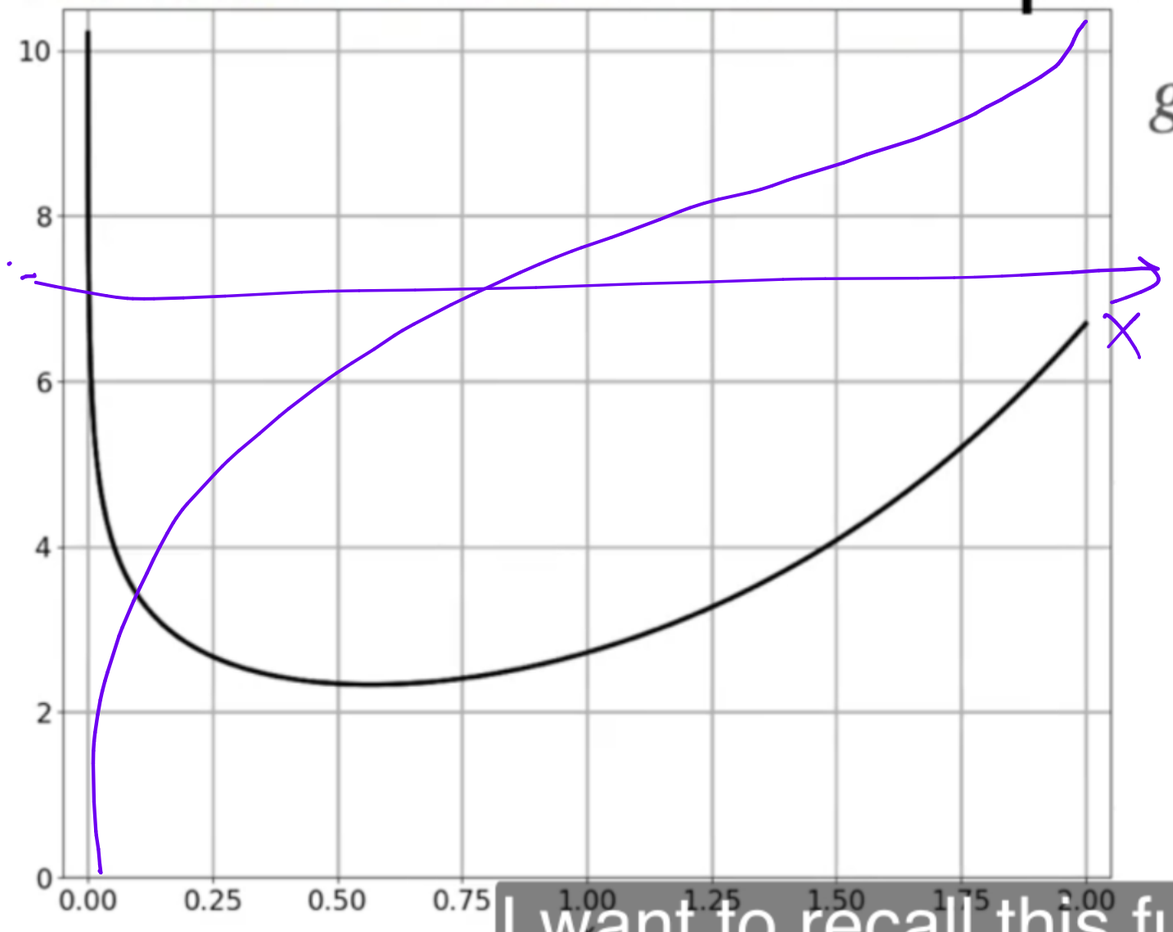
$$x_{k+1} = x_k - \frac{g'(x_k)}{(g'(x_k))'}$$

3) Repeat 2) until you find the candidate for minimum.

DeepLearning.AI

Newton's Method for Optimization

Newton's Method for Optimization



$f(x)$

$g(x) = e^x - \log(x)$ | $g'(x) = e^x - 1/x$

Minimum: $x^* = 0.5671$

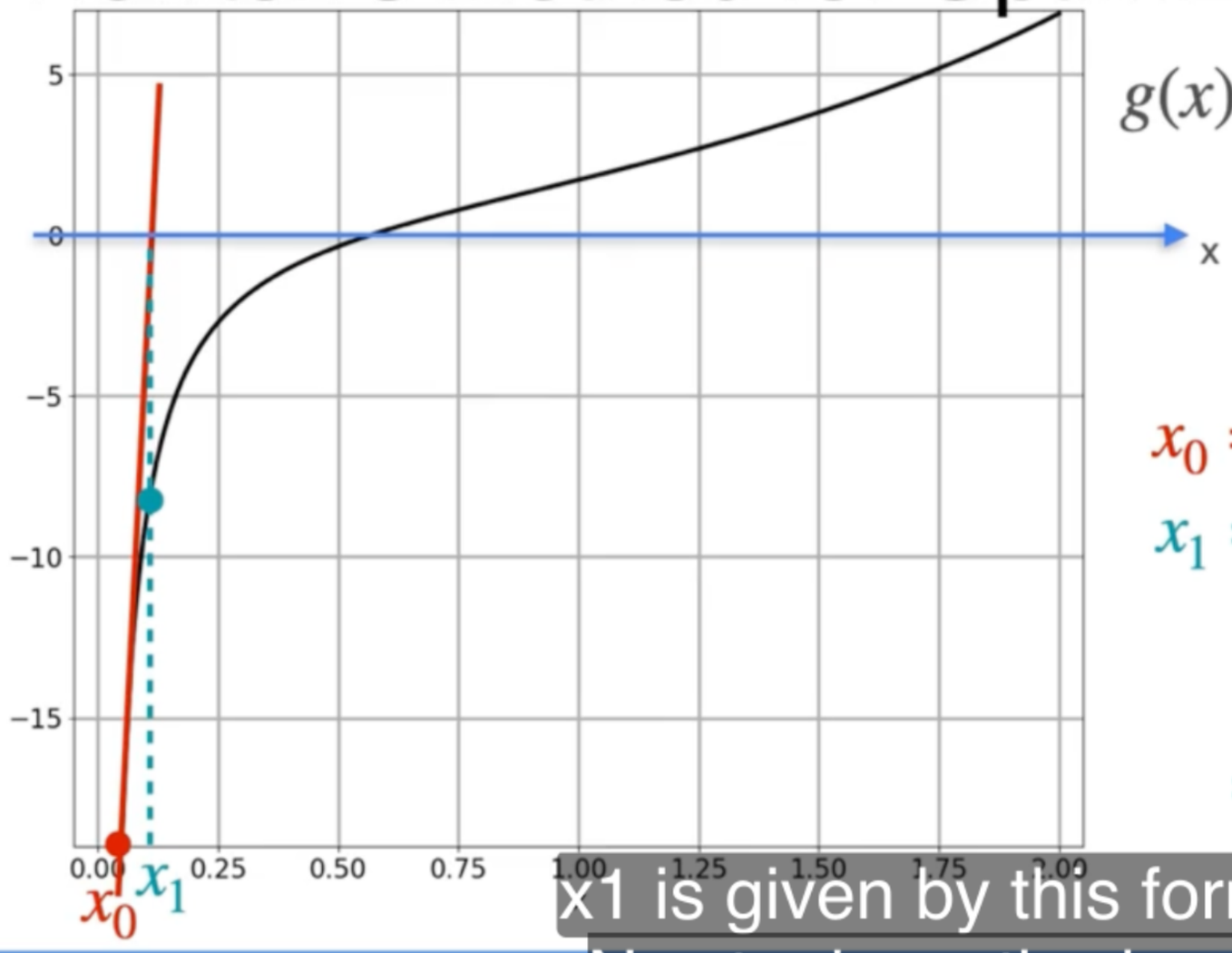
$(g'(x))' = e^x + \frac{1}{x^2}$

$f'(x)$

I want to recall this function you saw
in Week 2 when you learn about gradient

DeepLearning.AI

Newton's Method for Optimization



$$g(x) = e^x - \log(x) \qquad \overbrace{g'(x) = e^x - 1/x}^{f(x)}$$

Minimum: $x^* = 0.5671$

$$(g'(x))' = e^x + \frac{1}{x^2}$$

$$x_0 = 0.05 \qquad \overbrace{\quad}^{f'(x)}$$

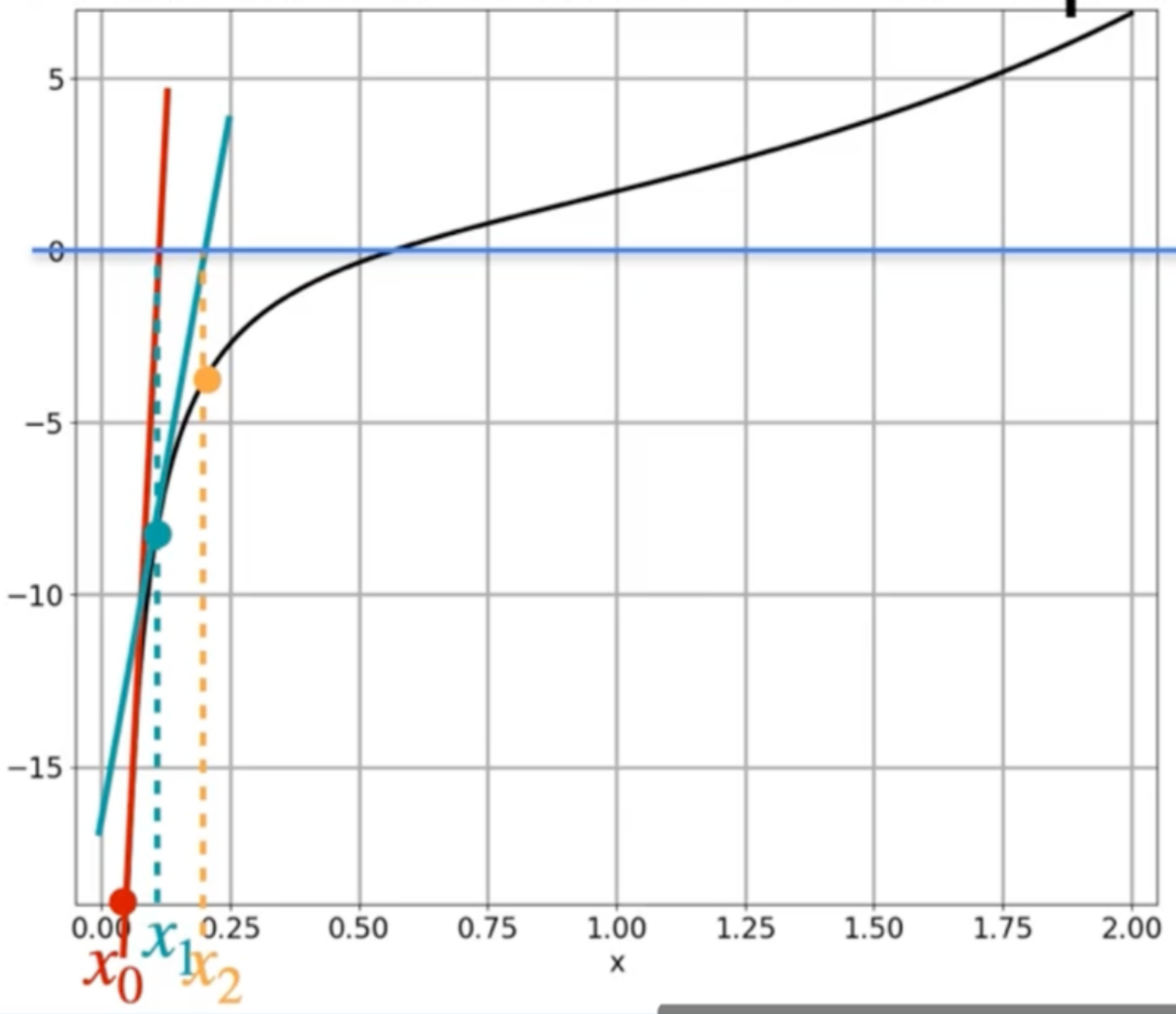
$$x_1 = x_0 - \frac{g'(x_0)}{(g'(x_0))'}$$

$$= 0.05 - \frac{\left(e^{0.05} - \frac{1}{0.05}\right)}{\left(e^{0.05} + \frac{1}{0.05^2}\right)} \Rightarrow 0.097$$

x1 is given by this formula using Newton's method we evaluate

DeepLearning.AI

Newton's Method for Optimization



$$g(x) = e^x - \log(x) \qquad g'(x) = e^x - 1/x$$

Minimum: $x^* = 0.5671$

$$(g'(x))' = e^x + \frac{1}{x^2}$$

$$x_1 = 0.097$$

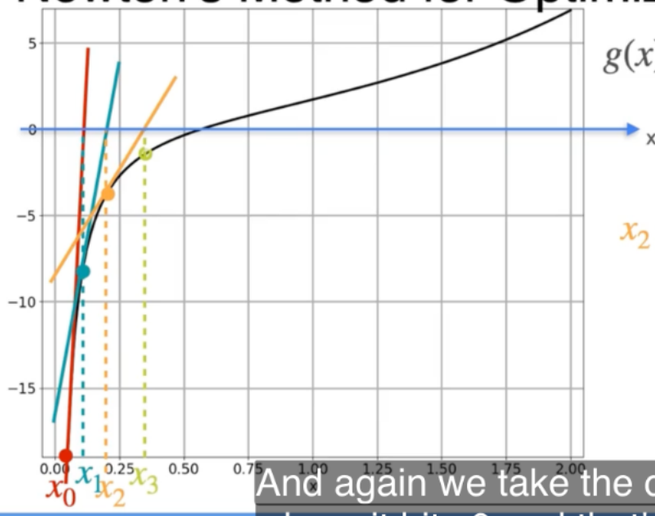
$$x_2 = x_1 - \frac{g'(x_1)}{(g'(x_1))'}$$

$$= 0.097 - \frac{\left(e^{0.097} - \frac{1}{0.097}\right)}{\left(e^{0.097} + \frac{1}{0.097^2}\right)} = 0.183$$

By plugging in these values.

DeepLearning.AI

Newton's Method for Optimization



$$g(x) = e^x - \log(x) \qquad g'(x) = e^x - 1/x$$

Minimum: $x^* = 0.5671$

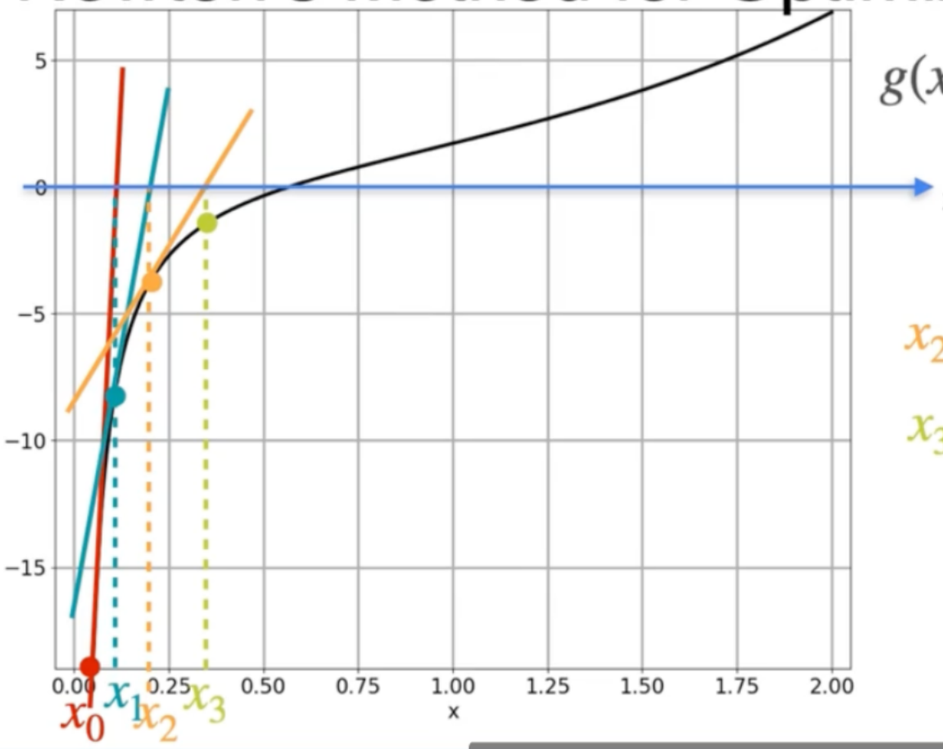
$$(g'(x))' = e^x + \frac{1}{x^2}$$

$x_2 = 0.183$

And again we take the derivative, see where it hits 0 and that's our point x3.

DeepLearning.AI

Newton's Method for Optimization



$$g(x) = e^x - \log(x) \qquad g'(x) = e^x - 1/x$$

Minimum: $x^* = 0.5671$

$$(g'(x))' = e^x + \frac{1}{x^2}$$

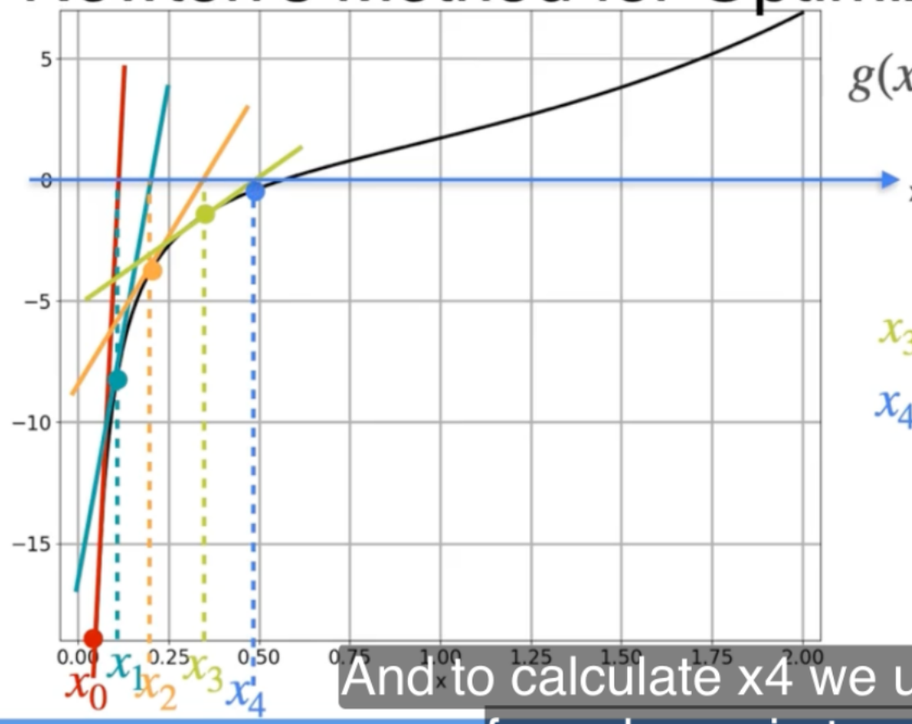
$x_2 = 0.183$

$$x_3 = x_2 - \frac{g'(x_2)}{(g'(x_2))'}$$

$$= 0.183 - \frac{\left(e^{0.183} - \frac{1}{0.183}\right)}{\left(e^{0.183} + \frac{1}{0.183^2}\right)} = 0.320$$

And the value for x3 is 0.320.

Newton's Method for Optimization



$$g(x) = e^x - \log(x) \qquad g'(x) = e^x - 1/x$$

Minimum: $x^* = 0.5671$

$$(g'(x))' = e^x + \frac{1}{x^2}$$

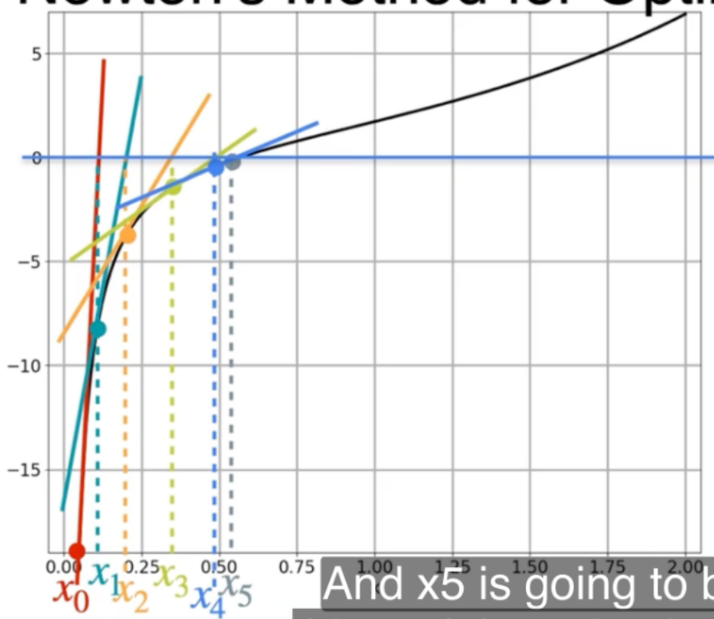
$x_3 = 0.320$

$$x_4 = x_3 - \frac{g'(x_3)}{(g'(x_3))'}$$

$$= 0.320 - \frac{\left(e^{0.320} - \frac{1}{0.320}\right)}{\left(e^{0.320} + \frac{1}{0.320^2}\right)} = 0.477$$

And to calculate x4 we use the iterative formula again to get 0.477

Newton's Method for Optimization



$$g(x) = e^x - \log(x) \qquad g'(x) = e^x - 1/x$$

Minimum: $x^* = 0.5671$

$$(g'(x))' = e^x + \frac{1}{x^2}$$

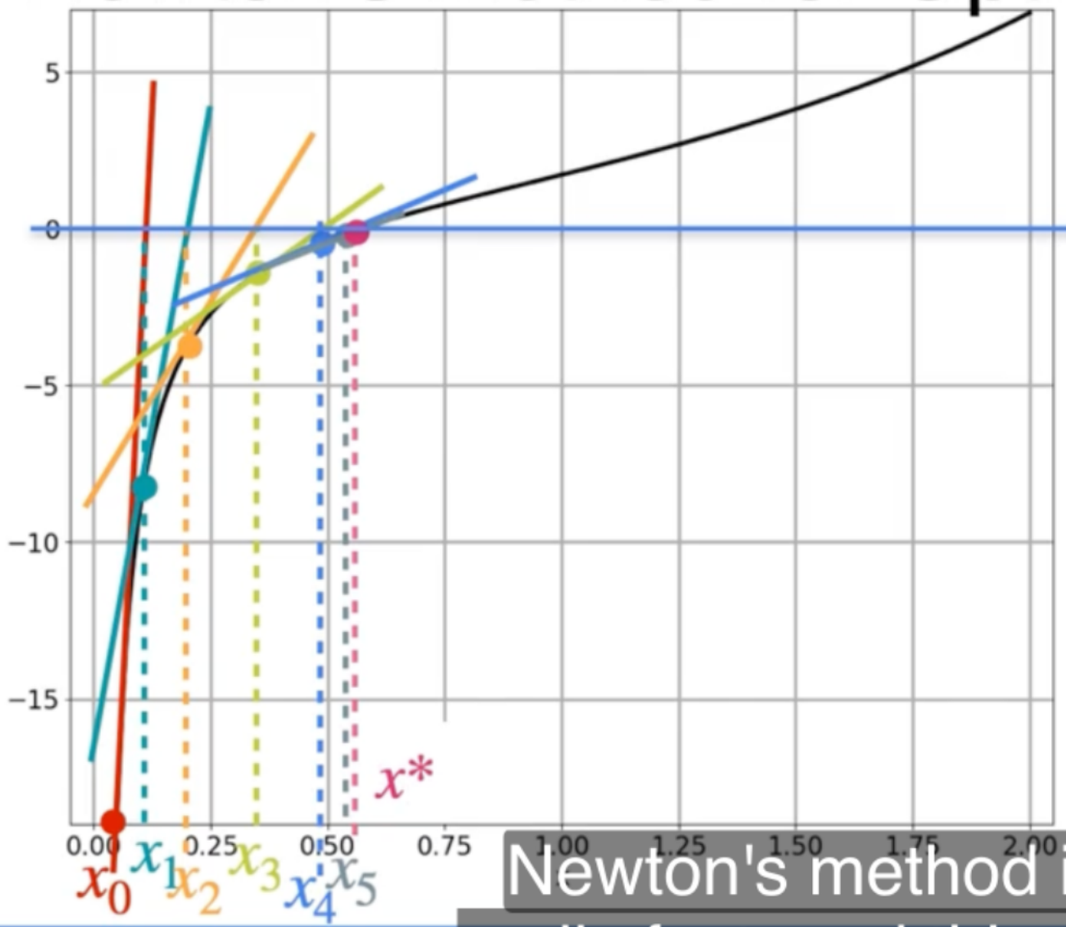
$x_4 = 0.477$

$$x_5 = x_4 - \frac{g'(x_4)}{(g'(x_4))'}$$

$$= 0.447 - \frac{\left(e^{0.447} - \frac{1}{0.447}\right)}{\left(e^{0.447} + \frac{1}{0.447^2}\right)} = 0.558$$

And x5 is going to be calculated like this and the value is going to be 0.558.

Newton's Method for Optimization



$$g(x) = e^x - \log(x) \qquad g'(x) = e^x - 1/x$$

Minimum: $x^* = 0.567$

$$(g'(x))' = e^x + \frac{1}{x^2}$$

$$x_5 = 0.558$$

$$x^* = x_5 - \frac{g'(x_5)}{(g'(x_5))'}$$

$$= 0.558 - \frac{\left(e^{0.558} - \frac{1}{0.558}\right)}{\left(e^{0.558} + \frac{1}{0.558^2}\right)} = 0.567$$

Newton's method is actually really, really fast, and this example shows it.

DeepLearning.AI