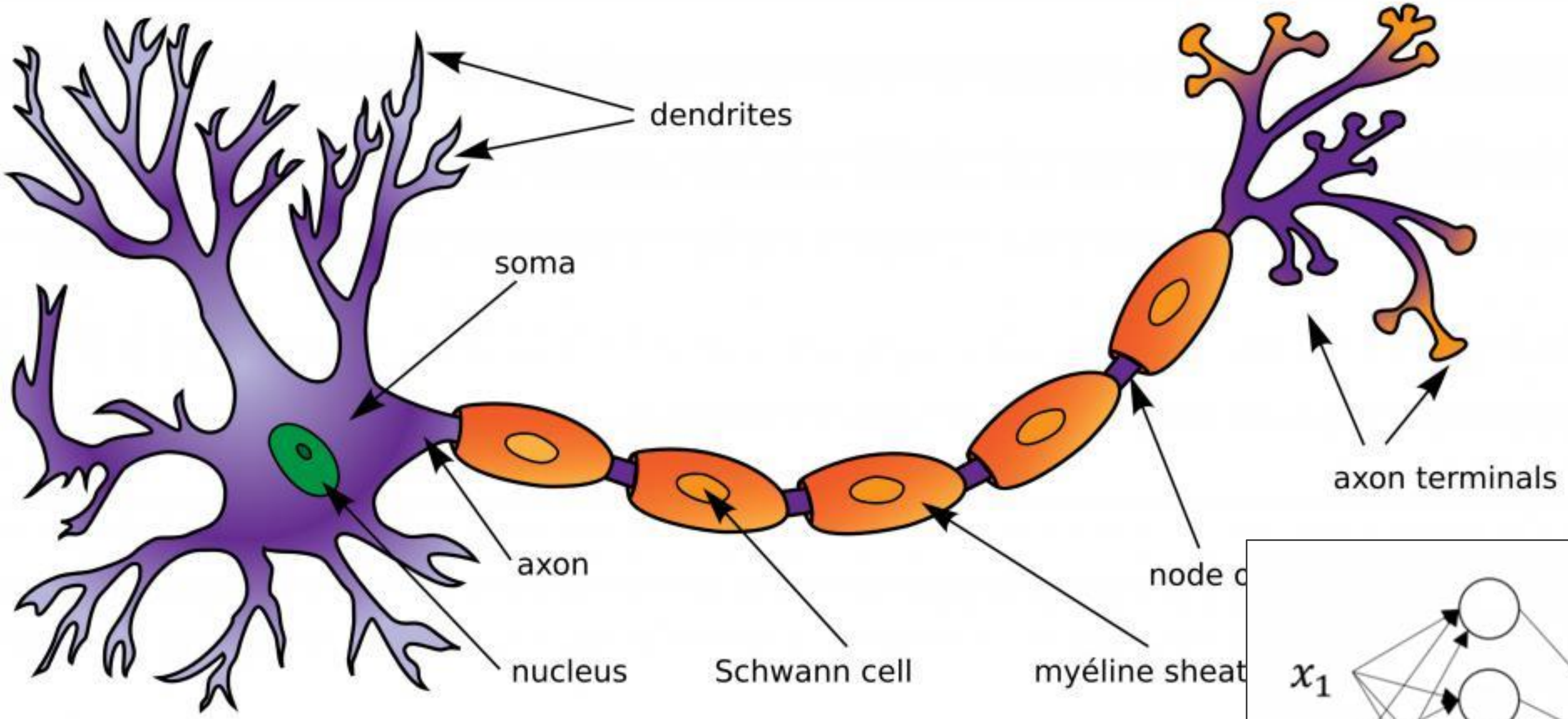
Shallow Neural Networks

= 1 hidden layer

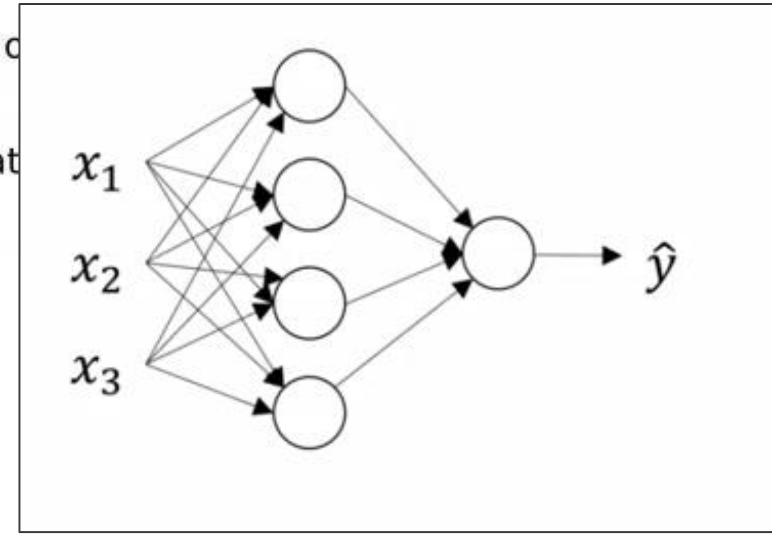
2021년 3월 20일 토요일 - 발표자 : 최윤정
20 Mar 2021 Sat- Speaker : Yoon Choi

NEURON

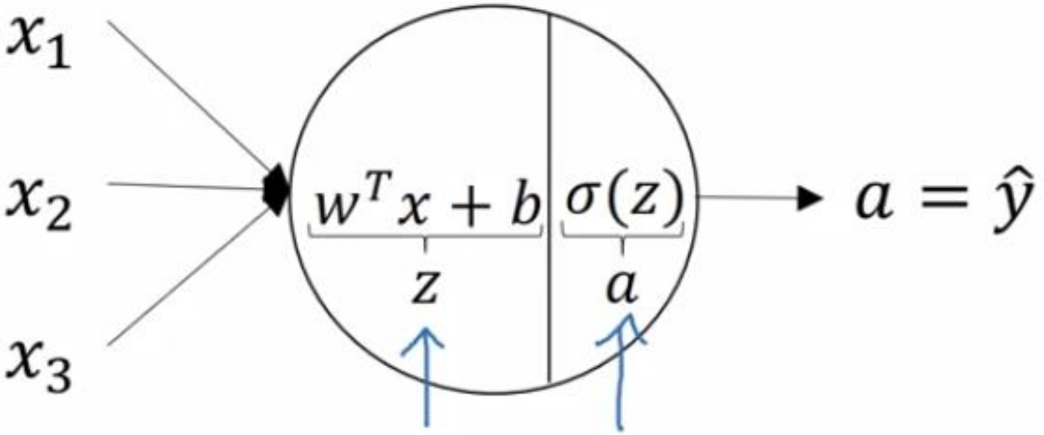


$$z = w^T x + b$$

$$a = \sigma(z)$$

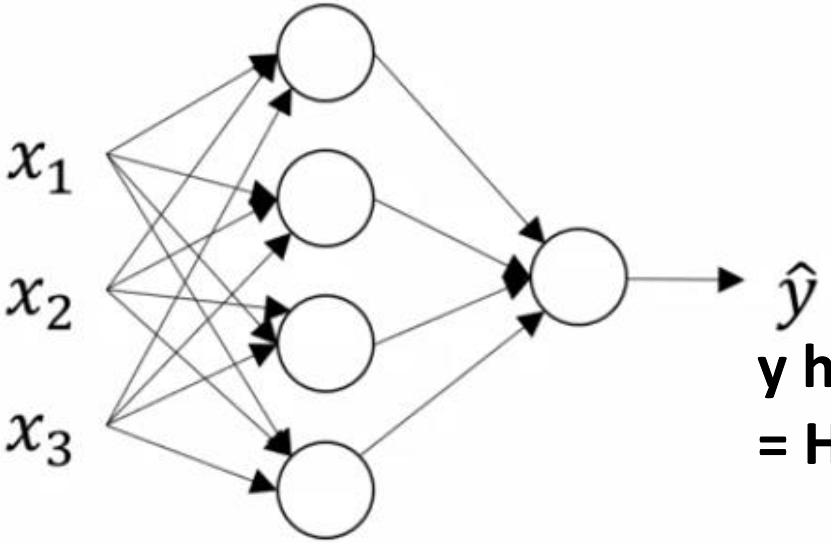


Neural Network Representation



$z = w^T x + b$

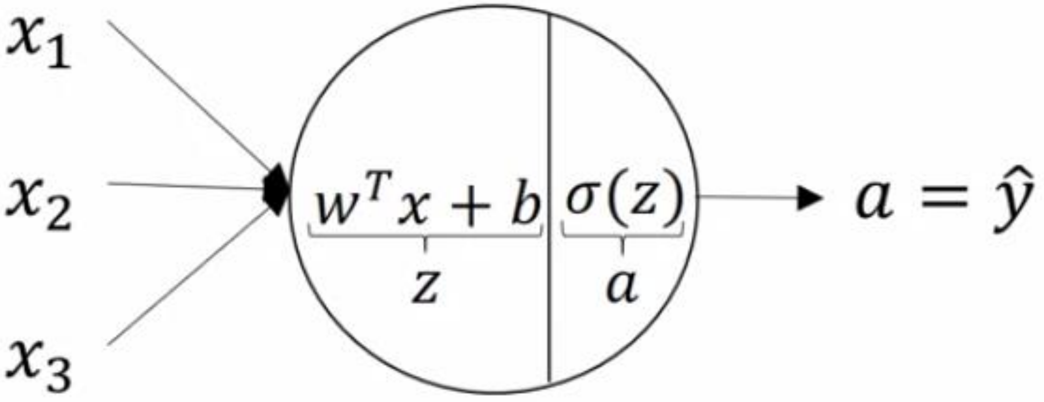
$a = \sigma(z)$



y hat
= Hypothesis

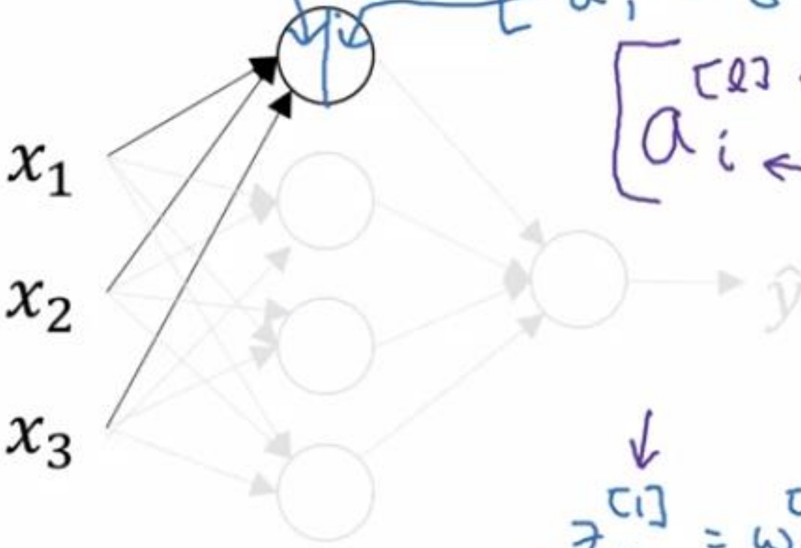
Andrew Ng

Neural Network Representation



$$z = w^T x + b$$

$$a = \sigma(z)$$



$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]}$$

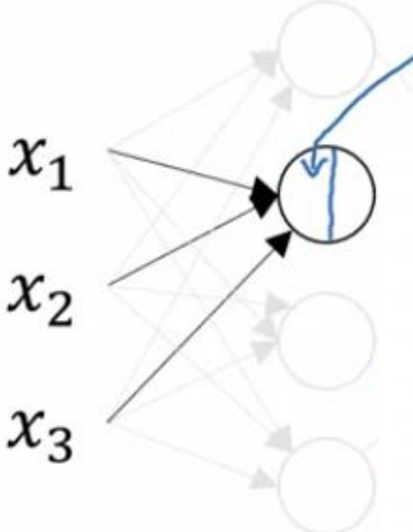
$$a_1^{[1]} = \sigma(z_1^{[1]})$$

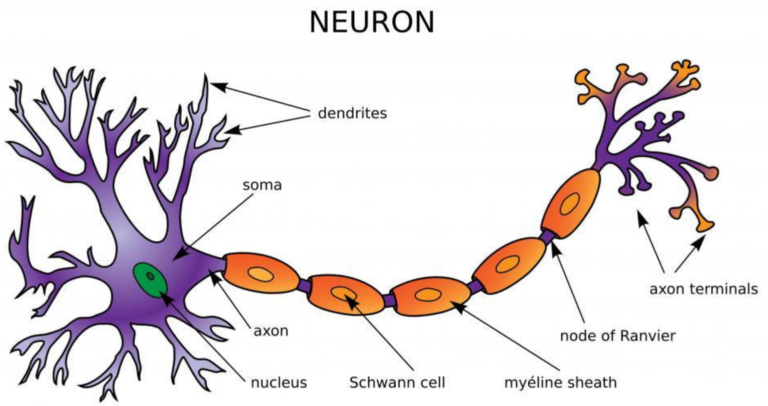
$a^{[l]}$ ← layer

a_i ← node in layer.

$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]}$$

$$a_2^{[1]} = \sigma(z_2^{[1]})$$



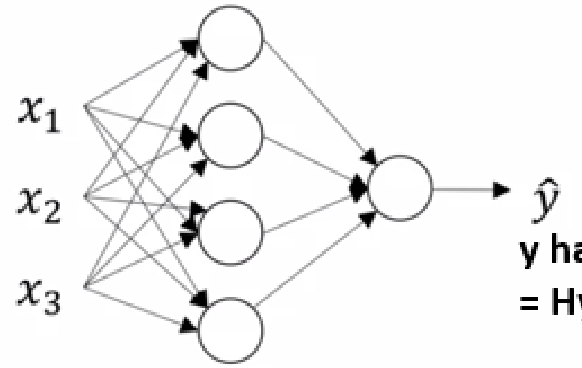


cost function of Classification

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log (1-h_\theta(x^{(i)}))]$$

cost function of Neural Network

$$\boxed{\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = -(y^{(i)} \log(\hat{y}^{(i)}) + (1-y^{(i)}) \log(1-\hat{y}^{(i)}))}$$



y hat
= Hypothesis

$$\boxed{\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)}) = -(y^{(i)} \log(\hat{y}^{(i)}) + (1-y^{(i)}) \log(1-\hat{y}^{(i)}))}$$

Gradient descent for neural networks

$\boxed{\text{Parameters:}}$ $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}$ $\qquad n_x = n^{[0]}, n^{[1]}, \underline{n^{[2]} = 1}$

$(n^{[1]}, n^{[0]}) \quad (n^{[1]}, 1) \quad (n^{[2]}, n^{[1]}) \quad (n^{[2]}, 1)$

$\boxed{\text{Cost function:}}$ $J(W^{[1]}, b^{[1]}, \underline{W}^{[2]}, \underline{b}^{[2]}) = \frac{1}{m} \sum_{i=1}^{n} \mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)})$

$\hat{y} = a^{[2]}$

$\boxed{\text{Gradient Descent:}}$ **Gradient of the Cost function을 구하기위해**
Back propagation 사용

→ Repeat {

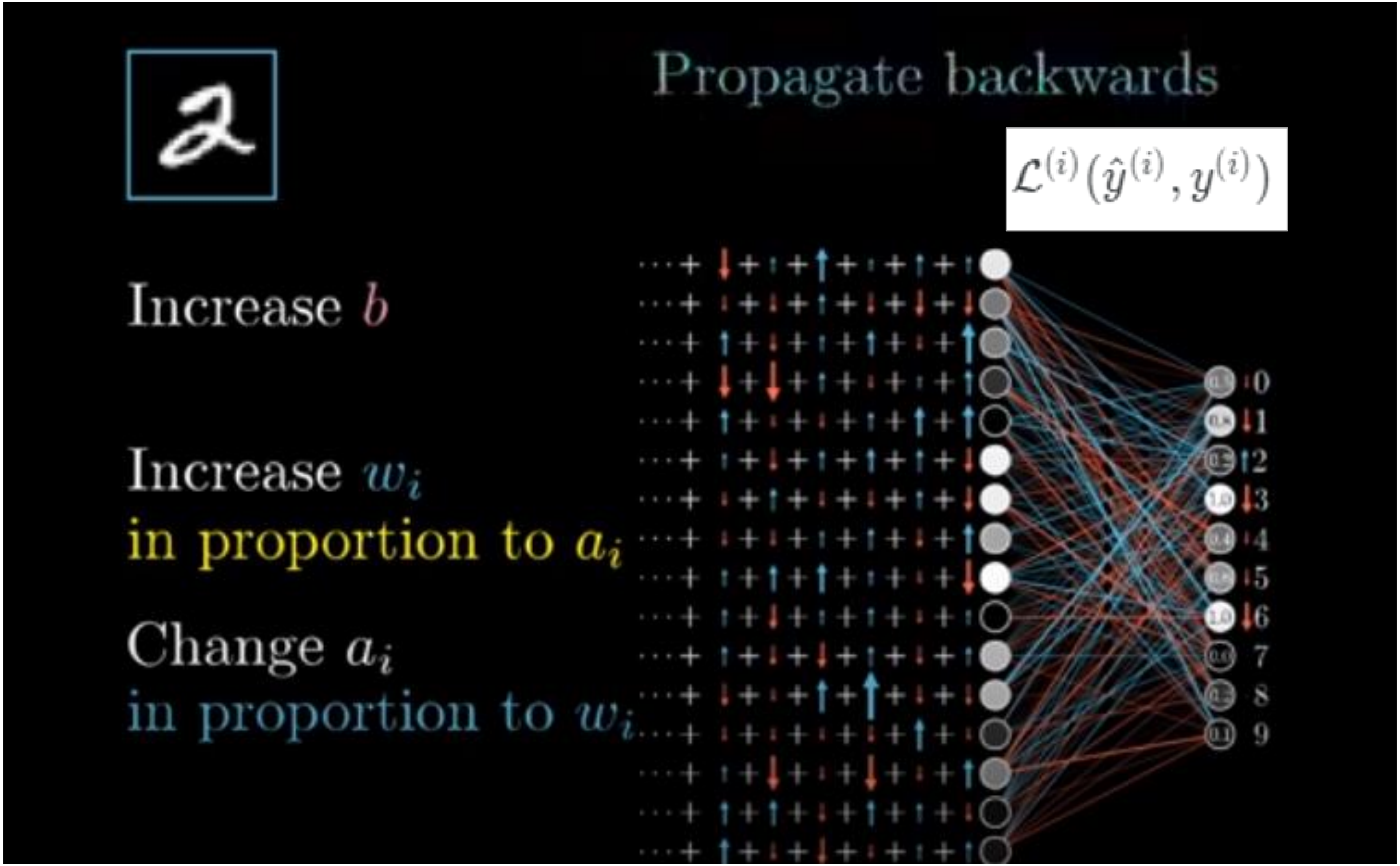
Compute predicts $(\hat{y}^{(i)}, i = 1 \ldots, m)$

$dW^{[1]} = \frac{\partial J}{\partial W^{[1]}}, \quad db^{[1]} = \frac{\partial J}{\partial b^{[1]}}, \ldots$

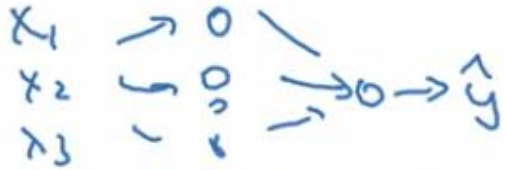
$W^{[1]} := W^{[1]} - \alpha \, dW^{[1]}$

$b^{[1]} := b^{[1]} - \alpha \, db^{[1]}$

$b^{[2]} := \ldots \quad b^{[2]} \ldots$


Propagate backwards
$\mathcal{L}^{(i)}(\hat{y}^{(i)}, y^{(i)})$
Increase b
Increase w_i
in proportion to a_i
Change a_i
in proportion to w_i
0
1
2
3
4
5
6
7
8
9

Neural network gradients



$n_x = n^{[0]}$ $\quad n^{[1]}$ $\quad n^{[2]} = 1$

$W^{[2]}$ $\quad dW^{[2]}$

$b^{[2]}$ $\quad db^{[2]}$

x

$W^{[1]}$, $dW^{[1]}$ $\rightarrow$ $z^{[1]} = W^{[1]}x + b^{[1]}$ $\rightarrow$ $a^{[1]} = \sigma(z^{[1]})$ $\rightarrow$ $z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$ $\rightarrow$ $a^{[2]} = \sigma(z^{[2]})$ $\rightarrow$ $\mathcal{L}(a^{[2]}, y)$

$b^{[1]}$, $db^{[1]}$

$\rightarrow dz^{[1]} = W^{[2]T} dz^{[2]}$ $\quad da^{[1]} = \cdots$ $\quad dz^{[2]} = a^{[2]} - y$ $\quad da^{[2]}$

$* \, g^{[1]\prime}(z^{[1]})$

element wise product

$$\left[\begin{aligned} dW^{[2]} &= dz^{[2]} a^{[1]T} \\ db^{[2]} &= dz^{[2]} \end{aligned}\right.$$

$\rightarrow$ "$dw = dz \cdot x$"

$w^{[1]}: [\quad\quad]$

$W^{[2]}$ $\quad (n^{[2]}, n^{[1]})$

$\rightarrow z^{[2]}, dz^{[2]}$ $\quad (n^{[2]}, 1)$ $\quad - (1,1)$

$\rightarrow z^{[1]}, dz^{[1]}$ $\quad (n^{[1]}, 1)$

$$dz^{[1]} = \boxed{W^{[2]T} dz^{[2]}} * \boxed{g^{[1]\prime}(z^{[1]})}$$

$$(n^{[1]}, 1) = (n^{[1]}, n^{[2]}) \; (n^{[2]}, 1) * (n^{[1]}, 1)$$

foo $\quad W^{[1]}$

dfoo $\quad dW^{[1]}$

$$\left[\begin{aligned} dW^{[1]} &= dz^{[1]} \cdot x^T \\ db^{[1]} &= dz^{[1]} \end{aligned}\right.$$

$a^{[0]T}$

Andrew Ng

Summary of gradient descent

$$dz^{[2]} = a^{[2]} - y$$

$$dW^{[2]} = dz^{[2]} a^{[1]^T}$$

$$db^{[2]} = dz^{[2]}$$

$$\underset{(n^{[1]},1)}{dz^{[1]}} = W^{[2]T} dz^{[2]} * g^{[1]\prime}(\mathrm{z}^{[1]})$$

$$dW^{[1]} = dz^{[1]} x^T$$

$$db^{[1]} = dz^{[1]}$$

$$dZ^{[2]} = A^{[2]} - Y$$

$$J(\ldots) = \frac{1}{m} \sum_{i=1}^{n} \mathcal{L}(\hat{y}, y)$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]^T}$$

$$db^{[2]} = \frac{1}{m} np.sum(dZ^{[2]}, axis = 1, keepdims = True)$$

elementwise product

$$\underset{(n^{[1]},m)}{dZ^{[1]}} = \underbrace{W^{[2]T} dZ^{[2]}}_{(n^{[1]},m)} * \underbrace{g^{[1]\prime}(\mathrm{Z}^{[1]})}_{(n^{[1]},m)}$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

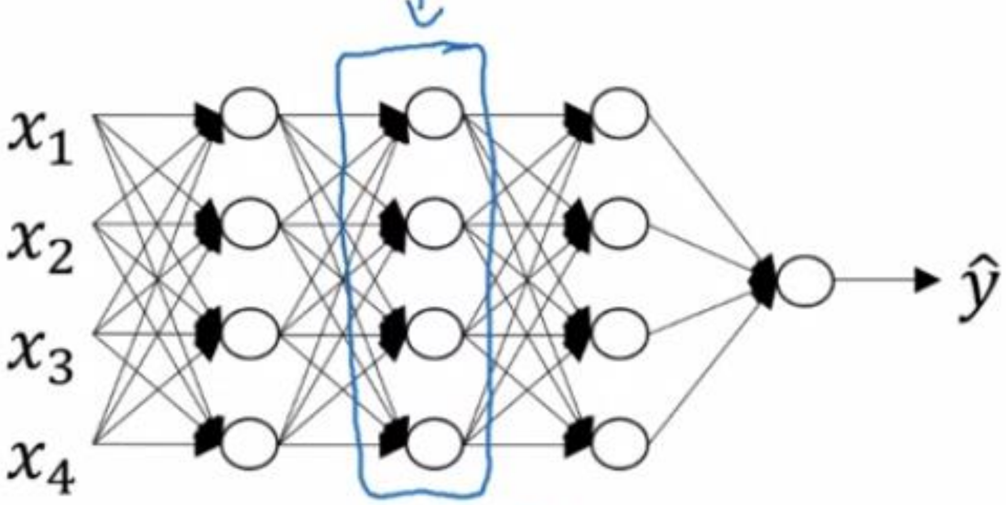
$$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$$

Andrew Ng

Deep Neural Networks

= more than 1 hidden layer

Forward and backward functions



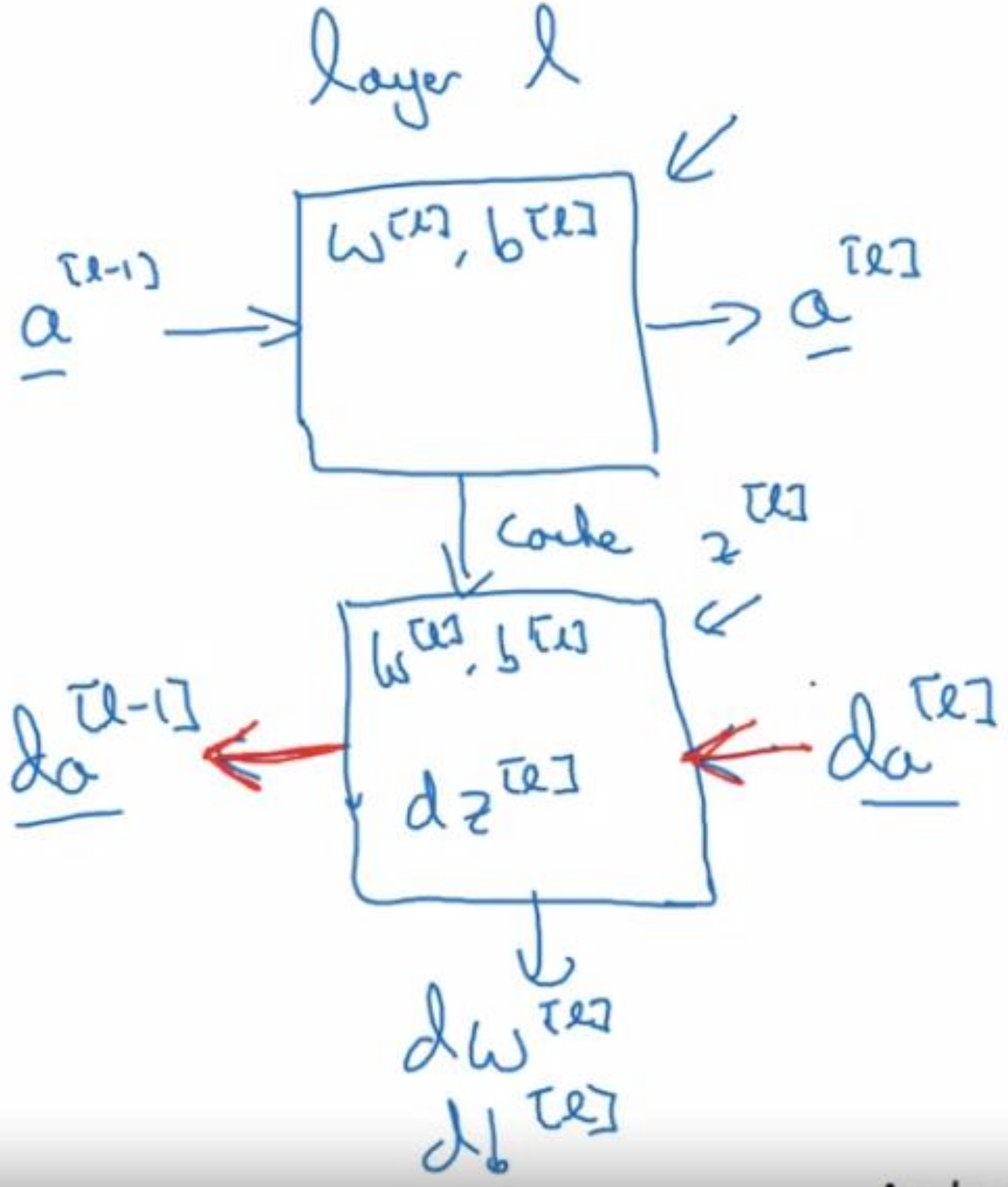
Layer l: $W^{[l]}, b^{[l]}$

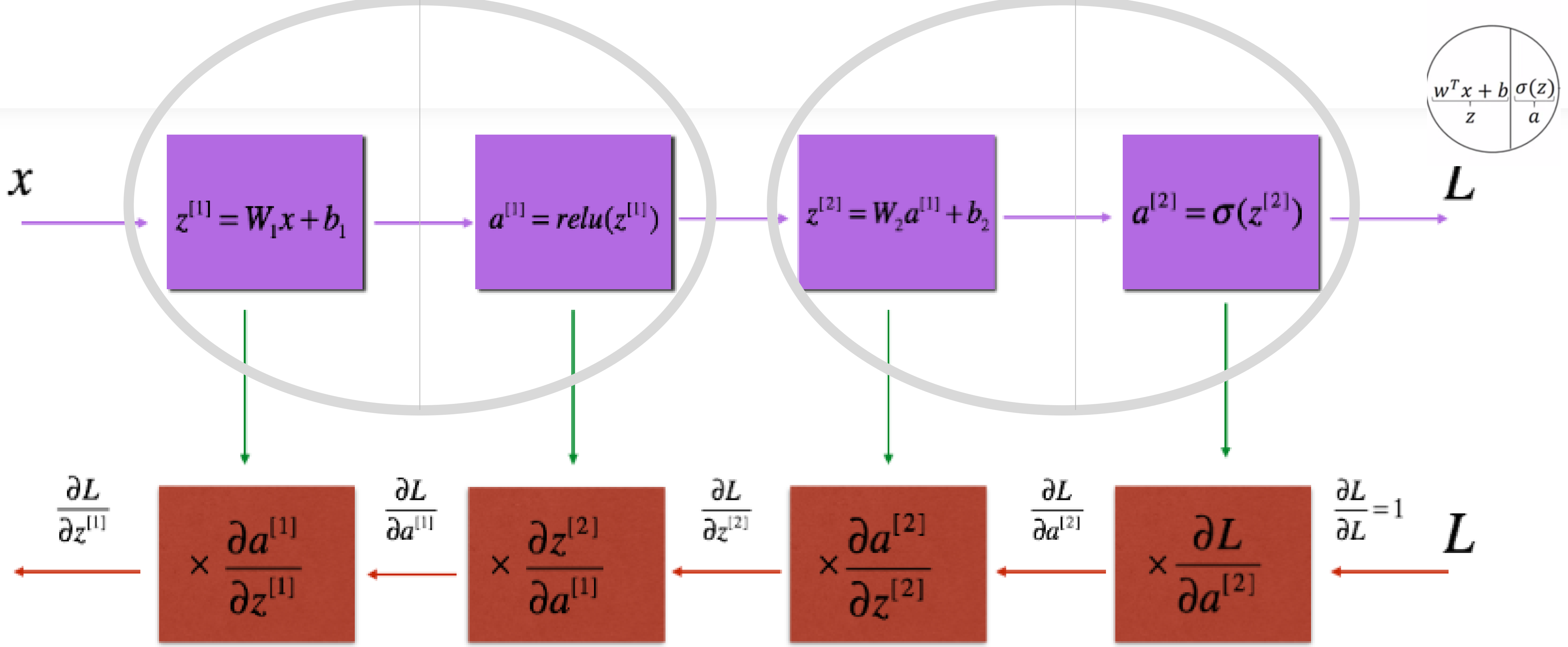
→ Forward: Input $a^{[l-1]}$, output $a^{[l]}$

$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$ cache $z^{[l]}$

$a^{[l]} = g^{[l]}(z^{[l]})$

→ Backward: Input $da^{[l]}$, cache($z^{[l]}$) output $da^{[l-1]}$, $dW^{[l]}$, $db^{[l]}$





Now, similar to forward propagation, you are going to build the backward propagation in three steps:

- LINEAR backward
- LINEAR -> ACTIVATION backward where ACTIVATION computes the derivative of either the ReLU or sigmoid activation
- [LINEAR -> RELU] × (L-1) -> LINEAR -> SIGMOID backward (whole model)

Backward propagation for layer l

$$dW^{[l]} = \frac{\partial \mathcal{J}}{\partial W^{[l]}} = \frac{1}{m} dZ^{[l]} A^{[l-1]T}$$

$$db^{[l]} = \frac{\partial \mathcal{J}}{\partial b^{[l]}} = \frac{1}{m} \sum_{i=1}^{m} dZ^{[l](i)}$$

$$dA^{[l-1]} = \frac{\partial \mathcal{L}}{\partial A^{[l-1]}} = W^{[l]T} dZ^{[l]}$$

→ Input $da^{[l]}$

→ Output $da^{[l-1]}, dW^{[l]}, db^{[l]}$

$$dz^{[l]} = da^{[l]} * g^{[l]'}(z^{[l]})$$

$$dW^{[l]} = dz^{[l]} \cdot a^{[l-1]}$$

$$db^{[l]} = dz^{[l]}$$

$$da^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$$

$$dz^{[l]} = W^{[l+1]T} dz^{[l+1]} * g^{[l]'}(z^{[l]})$$

$$dZ^{[l]} = dA^{[l]} * g^{[l]'}(Z^{[l]})$$

$$dW^{[l]} = \frac{1}{m} dZ^{[l]} \cdot A^{[l-1]T}$$

$$db^{[l]} = \frac{1}{m} \text{np.sum}(dZ^{[l]}, \text{axis}=1, \text{keepdims}=\text{True})$$

$$dA^{[l-1]} = W^{[l]T} \cdot dZ^{[l]}$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

Share

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

Hyperparameters: learning rate α

#iterations

#hidden layers L

#hidden units $n^{[1]}, n^{[2]}, \ldots$

choice of activation function

Later: Momentum, mini-batch size, regularizations ...

hyperparameter : parameter의 값이나 상태를 결정짓는 요소들

Andrew Ng

2:15 / 7:16

- [x] weight matrices $W^{[l]}$

This should not be selected

- [x] bias vectors $b^{[l]}$

This should not be selected

- [x] number of layers L in the neural network

Correct

- [x] size of the hidden layers $n^{[l]}$

Correct

- [x] learning rate α

Correct

- [x] number of iterations

Correct

- [x] activation values $a^{[l]}$

This should not be selected