One hidden layer
Neural Network

Backpropagation
intuition (Optional)

deeplearning.ai

Computing gradients

Logistic regression

x_1
x_2 $\rightarrow$ σ $\rightarrow$ $\hat{y} = a$
x_3

x_1
$x_2 \rightarrow$ $\circ$ $\circ$ $\rightarrow \circ \rightarrow \hat{y}$
x_3

$$x \searrow$$
$$w \rightarrow \boxed{z = w^T x + b} \rightarrow \boxed{a = \sigma(z)} \rightarrow \boxed{\mathcal{L}(a, y)}$$
$$b \nearrow$$

$dz = a - y$

$dw = dz \cdot x$

$db = dz$

$$dz = da \cdot g'(z)$$

$g(z) = \sigma(z)$

$$\frac{\partial \ell}{\partial z} = \frac{\partial \ell}{\partial a} \cdot \boxed{\frac{da}{dz}}$$

$"dz" = "da" \cdot$

$da = \frac{d}{da} \ell(a, y) = -y \log a - (1-y) \log(1-a)$

$$= -\frac{y}{a} + \frac{1-y}{1-a}$$

$\frac{d}{dz} g(z) = \underline{g'(z)}$

Andrew Ng
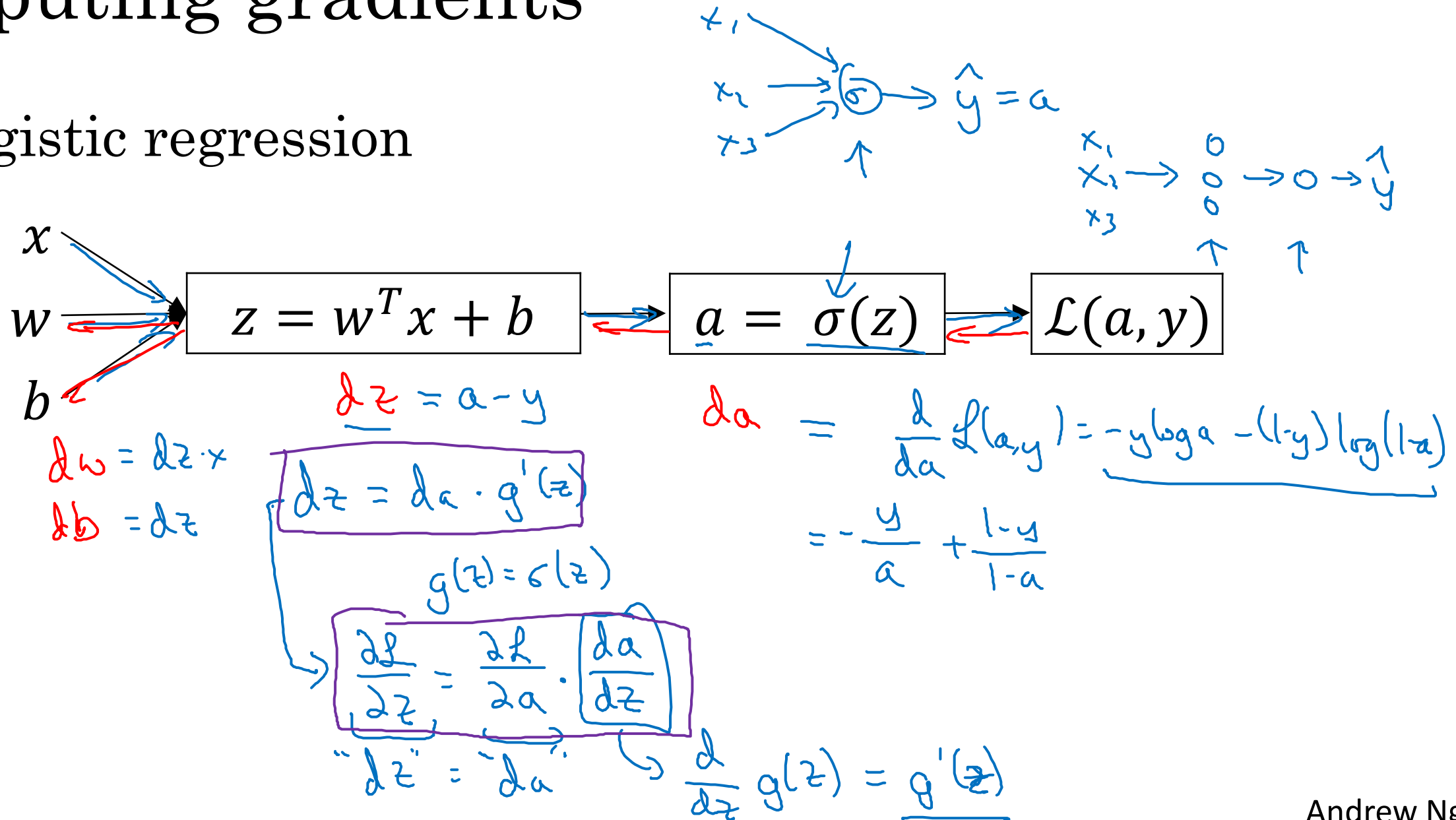
Neural network gradients

$x_1 \to \bigcirc$
$x_2 \to \bigcirc \to \bigcirc \to \hat{y}$
x_3

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

$W^{[2]}$ $dW^{[2]}$
$b^{[2]}$ $db^{[2]}$

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

x
$W^{[1]}$
$dw^{[1]}$
$b^{[1]}$
$db^{[1]}$

$dz^{[1]} = W^{[2]T} dz^{[2]}$

$da^{[1]} = \cdots$

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

$da^{[2]}$

$* g^{[1]'}(z^{[1]})$
element wise product

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

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

$``dw = dz \cdot x"$

$w^{[2]}$ $(n^{[2]}, n^{[1]})$

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

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

$W^{[2]}: [\underline{\quad\quad}]$

foo $W^{[1]}$ $dW^{[1]} = dz^{[1]} \cdot x^T$
$dfoo$ $dW^{[2]}$ $db^{[1]} = dz^{[1]}$ $a^{[0]T}$

$dz^{[1]} = \boxed{W^{[2]T} dz^{[2]}} * \boxed{g^{[1]'}(z^{[1]})}$

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

Andrew Ng
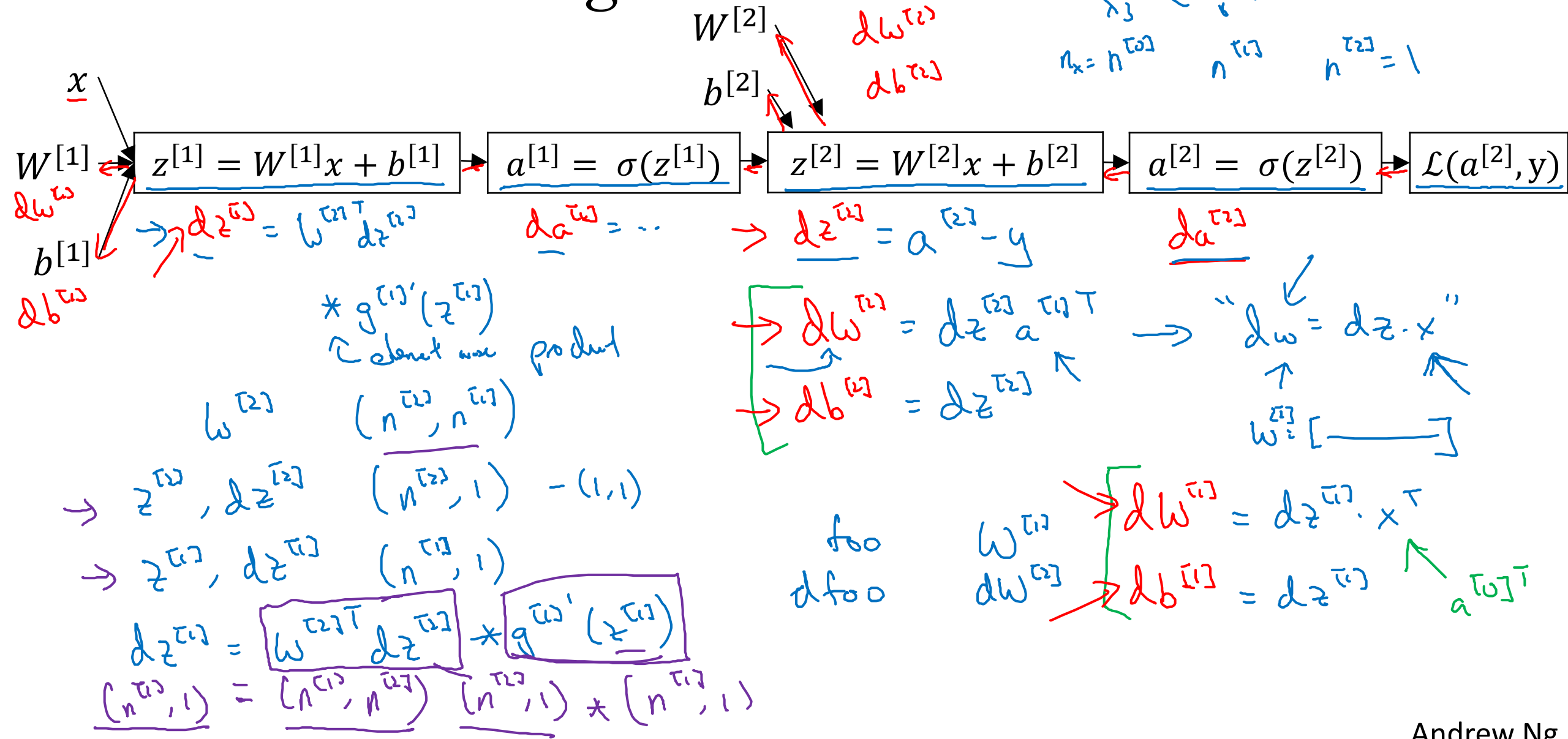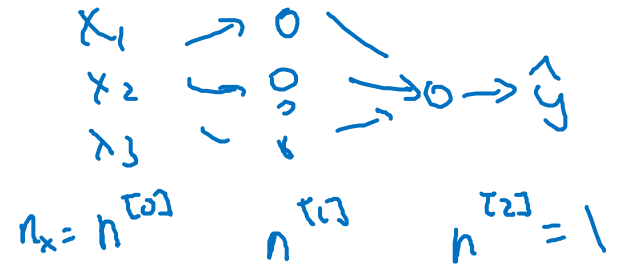
Summary of gradient descent

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

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

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

$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]'}(z^{[1]})$

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

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

Vectorized Implementation:

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

$a^{[1]} = g^{[1]}(z^{[1]})$

$Z^{[1]} = \begin{bmatrix} | & | & & | \\ z^{[1](1)} & z^{[1](2)} & \cdots & z^{[1](m)} \\ | & | & & | \end{bmatrix}$

$Z^{[1]} = W^{[1]} X + b^{[1]}$

$A^{[1]} = g^{[1]}(Z^{[1]})$

Summary of gradient descent

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

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

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

$$dz^{[1]} = W^{[2]T}dz^{[2]} * g^{[1]'}(z^{[1]})$$

$(n^{[1]}, 1)$

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

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

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

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

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

elementwise product

$$dZ^{[1]} = \underbrace{W^{[2]T}dZ^{[2]}}_{(n^{[1]}, m)} * \underbrace{g^{[1]'}(Z^{[1]})}_{(n^{[1]}, m)}$$

$(n^{[1]}, m)$

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

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

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

Andrew Ng